

**United States Patent** [19]  
**Caddy**

[11] Patent Number: 4,578,766

[45] **Date of Patent:** Mar. 25, 1986

[54] **COMPUTER-AIDED PROCESS FOR GENERATING CAMERA-READY GRAPHICAL ARTWORK**

**[75] Inventor: Michael J. Caddy, Doylestown, Pa.**

**[73] Assignee: The United States of America as represented by the Secretary of the Navy, Washington, D.C.**

[21] Appl. No.: 528,710

[22] Filed: Sep. 1, 1983

[51] Int. Cl.<sup>4</sup> ..... G06F 15/06; H04N 7/00

[52] U.S. Cl. .... 364/521; 364/300

[58] **Field of Search** ..... 364/521, 519, 520, 300,  
364/189, 518, 512; 340/728, 723

## [56] References Cited

## U.S. PATENT DOCUMENTS

3,579,194	5/1971	Weinblatt .....	364/300
4,148,066	4/1979	Saylor .....	364/521 X
4,475,161	10/1984	Stock .....	364/521
4,511,928	4/1985	Colomb .....	364/519 X
4,514,818	4/1985	Walker .....	364/521

## OTHER PUBLICATIONS

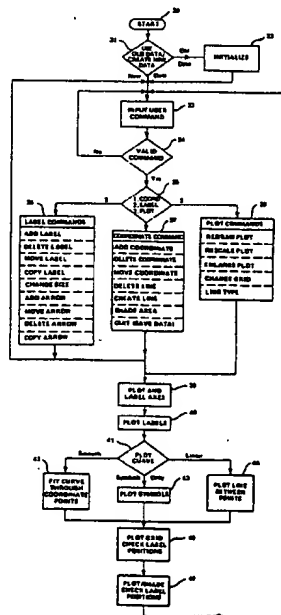
**A High Speed Graphics Display Controller, Electronic Product Design, Feb. 1982, pp. 43-47.**

**Primary Examiner**—Edward J. Wise  
**Attorney, Agent, or Firm**—Robert F. Beers; Henry Hansen; Armand M. Vozzo, Jr.

[57] **ABSTRACT**

A computer-aided process is disclosed for automatically generating a camera-ready hardcopy of a graphical plot upon command instructions inputted via a conventional storage tube graphics display terminal having an addressable cross-hair cursor and a keyboard. In accordance with an interactive graphics code or program, tabular data coordinates stored in computer file form are retrieved and plotted on appropriately titled and scaled axes with the plotted coordinates being interconnected along curves formed of a smooth or linear nature by interpolation. The graphical plot viewed on the display terminal is further enhanced by inclusion of labels, shaded areas, and reference symbols and characters prior to printing out the hardcopy of an associated graphics display unit coupled to the display terminal.

**5 Claims, 18 Drawing Figures**



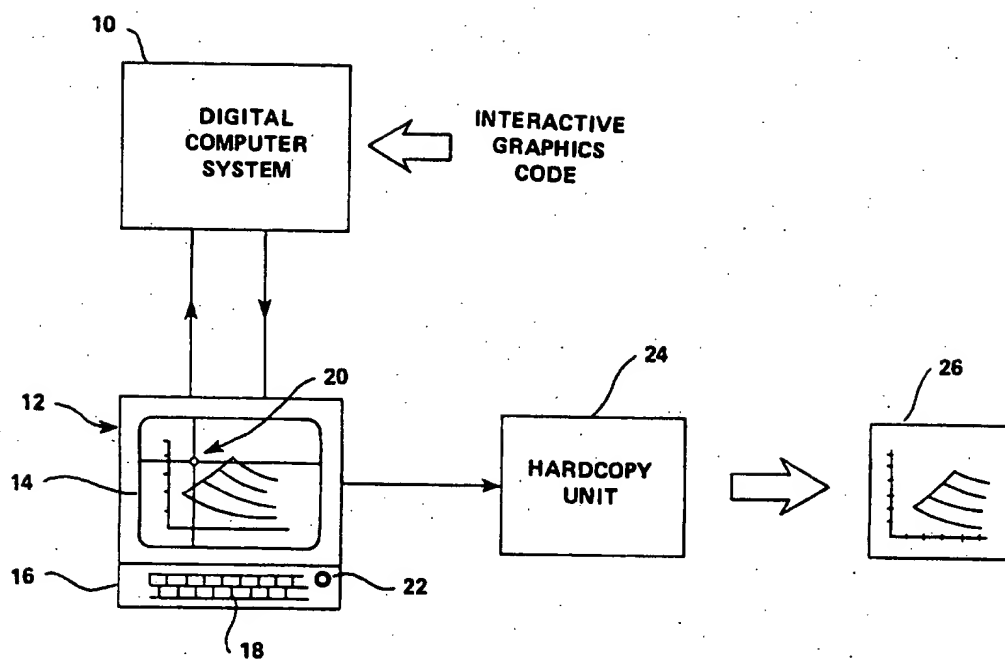


FIG. 1

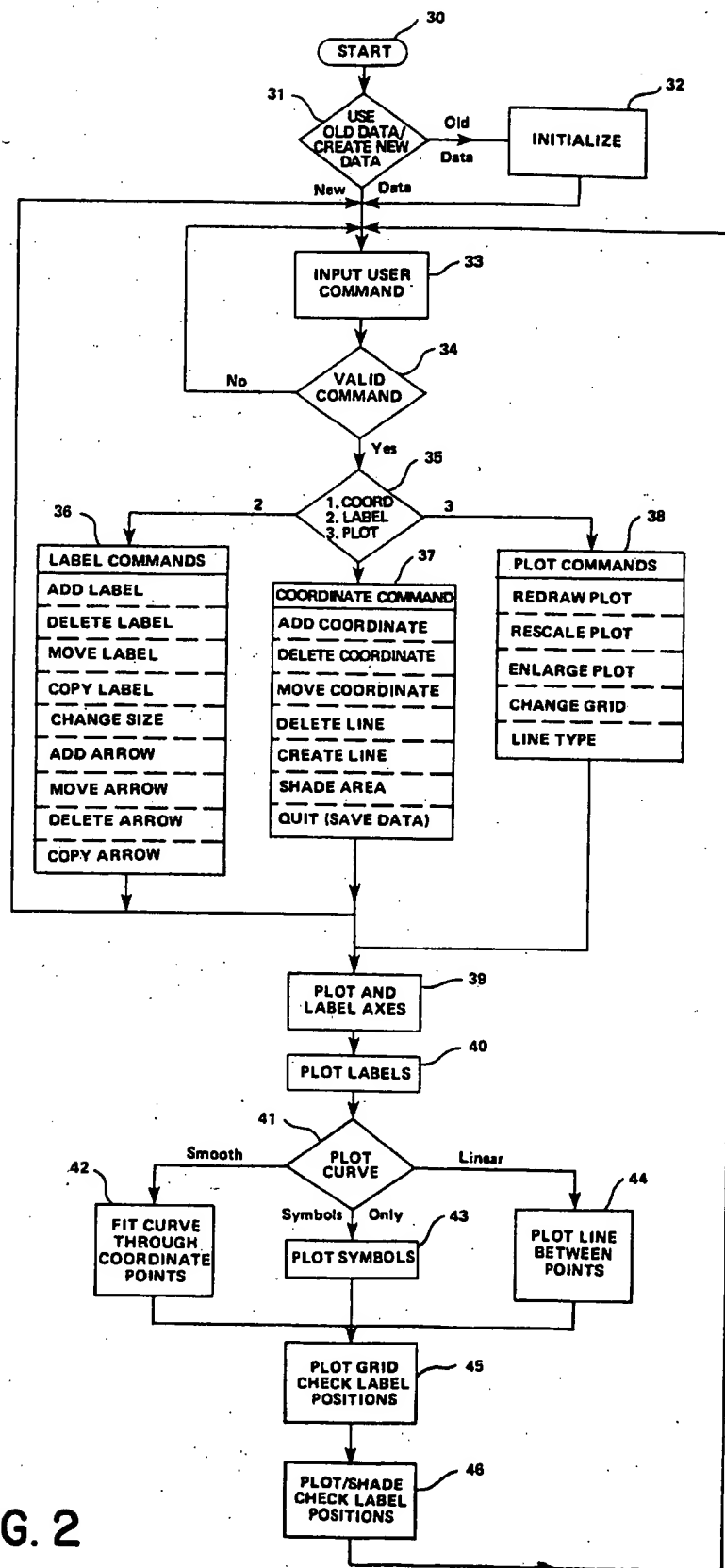


FIG. 2

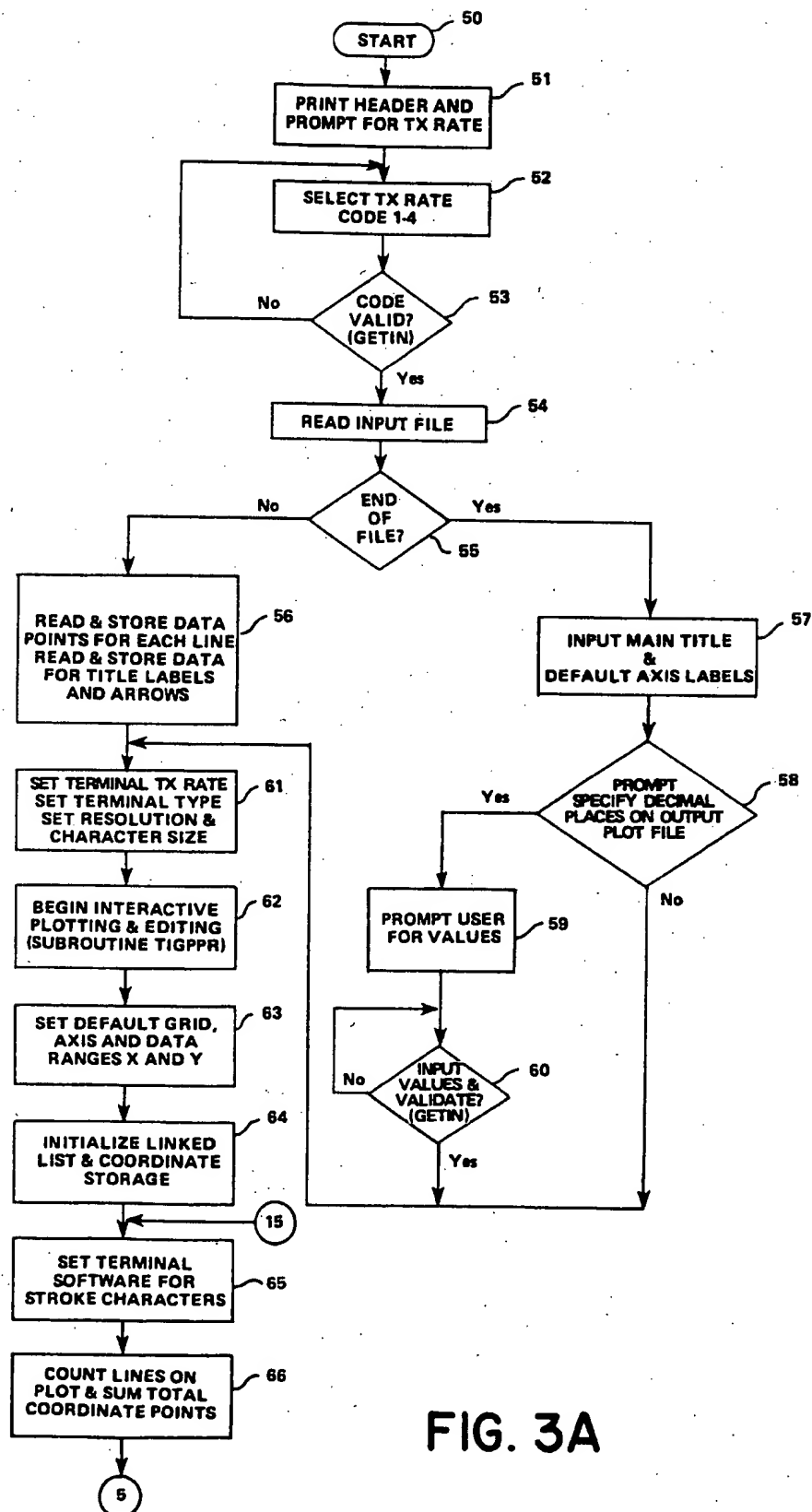


FIG. 3A

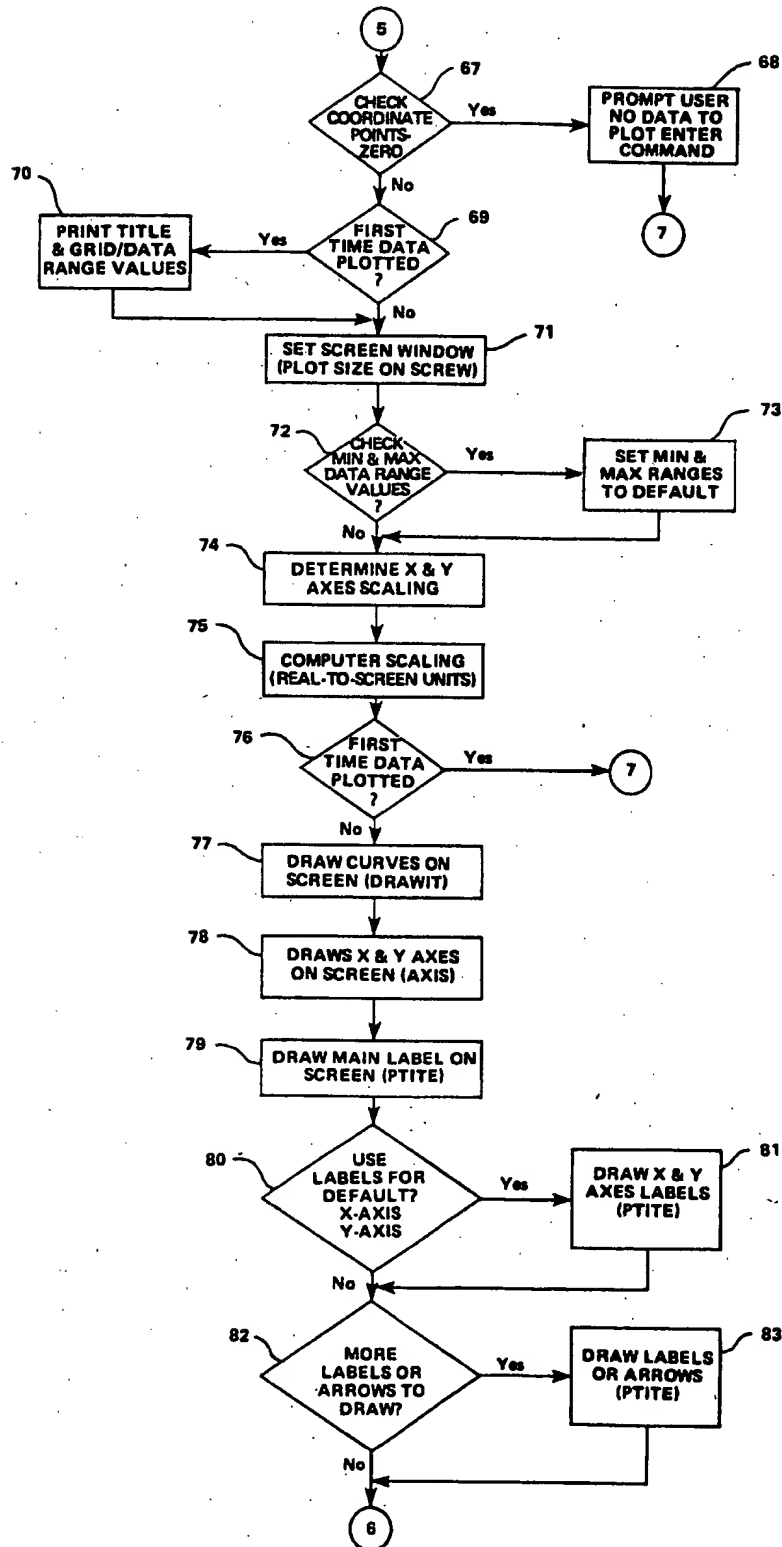


FIG. 3B

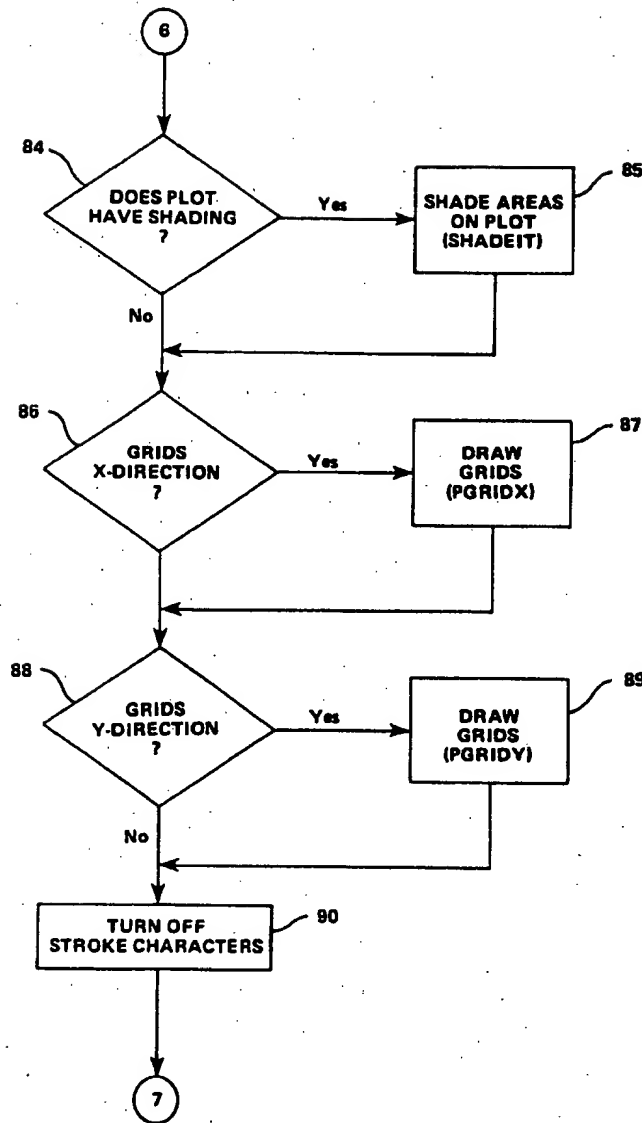


FIG. 3C

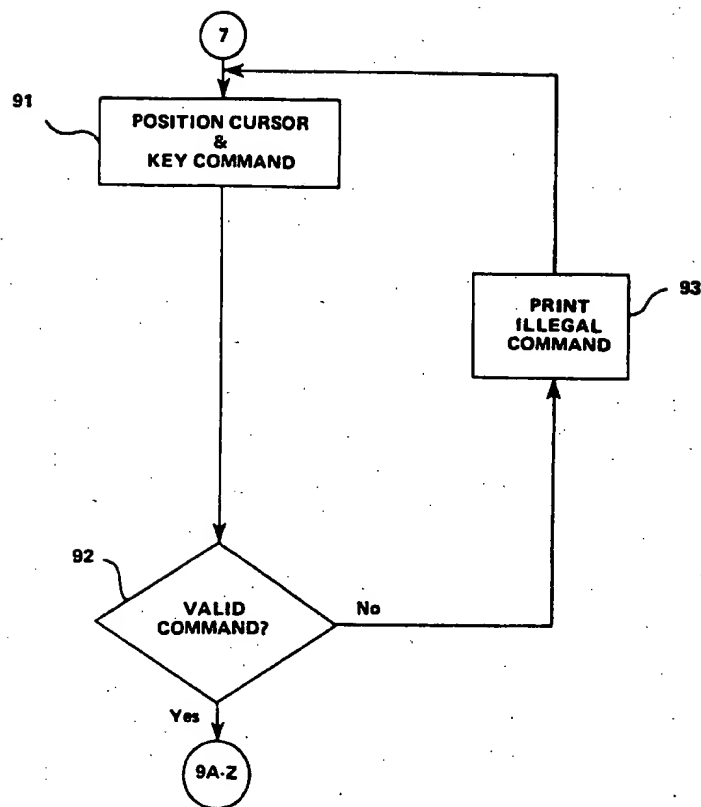


FIG. 3D

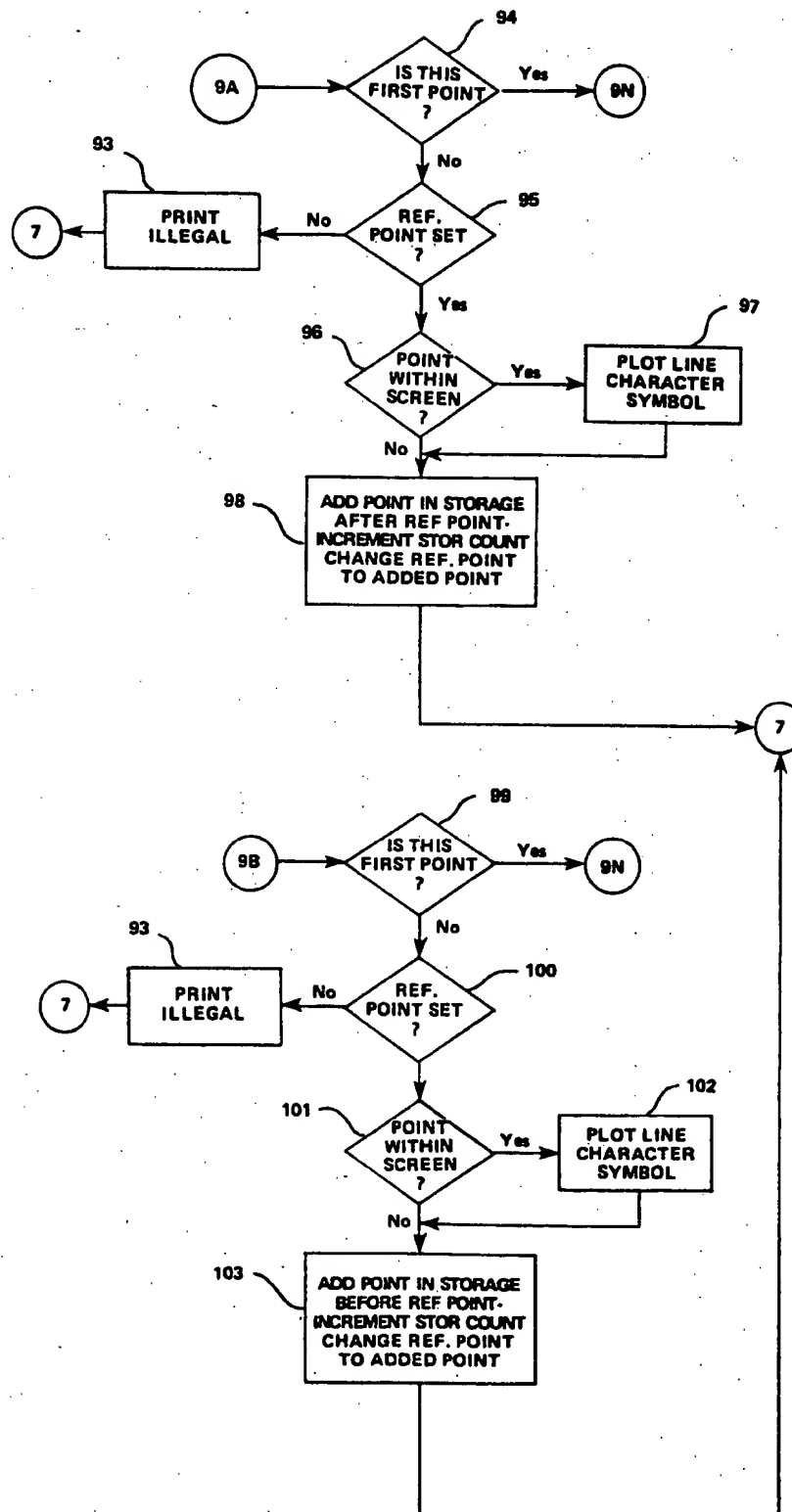


FIG. 3E



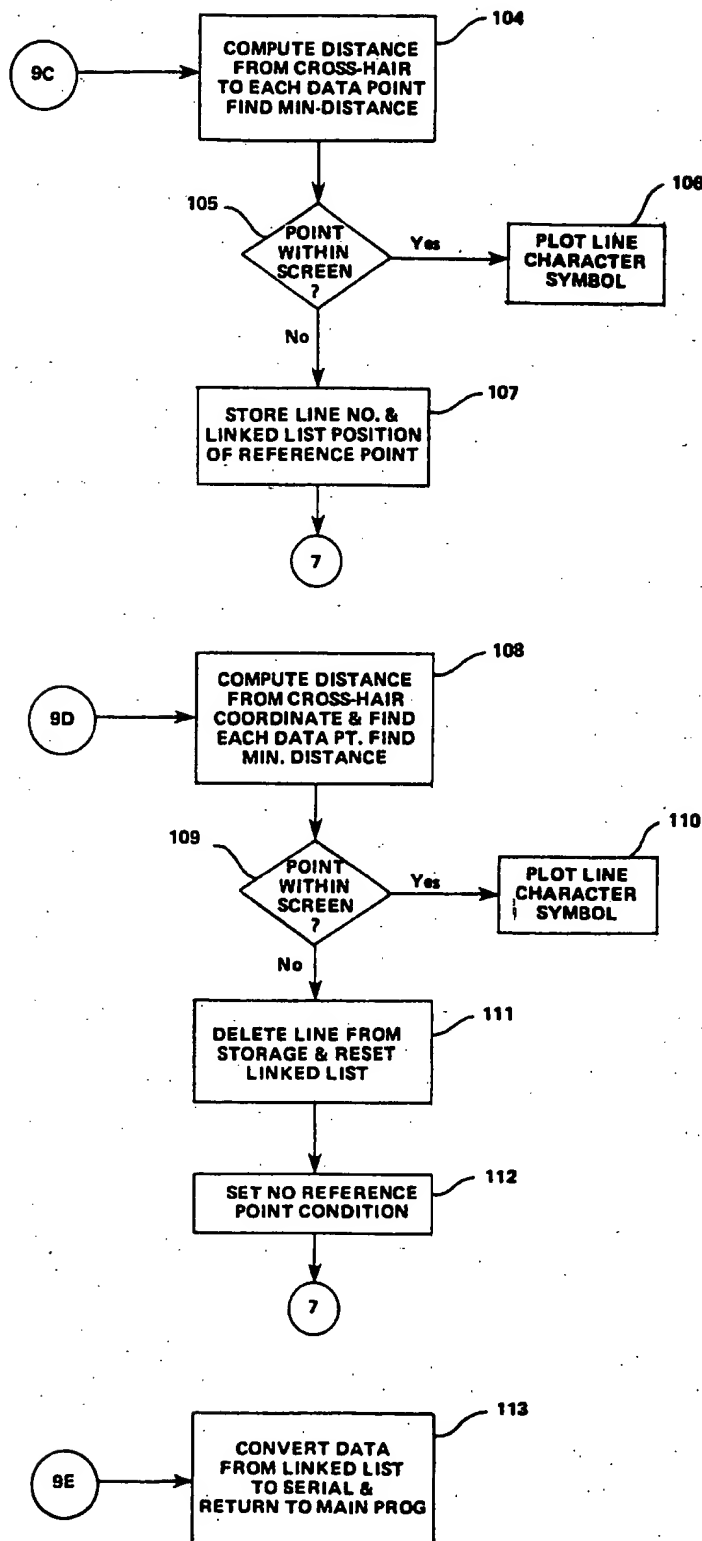


FIG. 3F

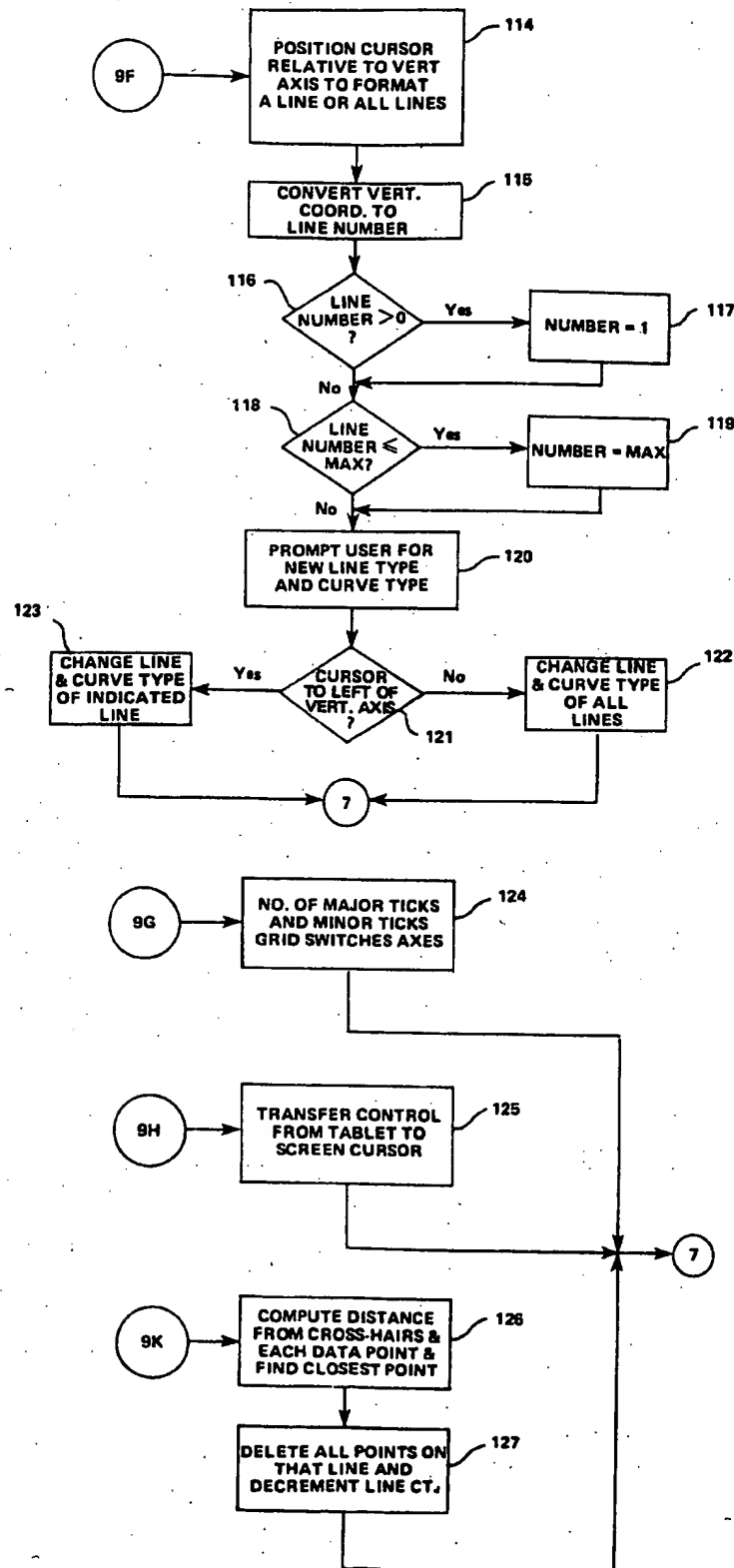


FIG. 3G

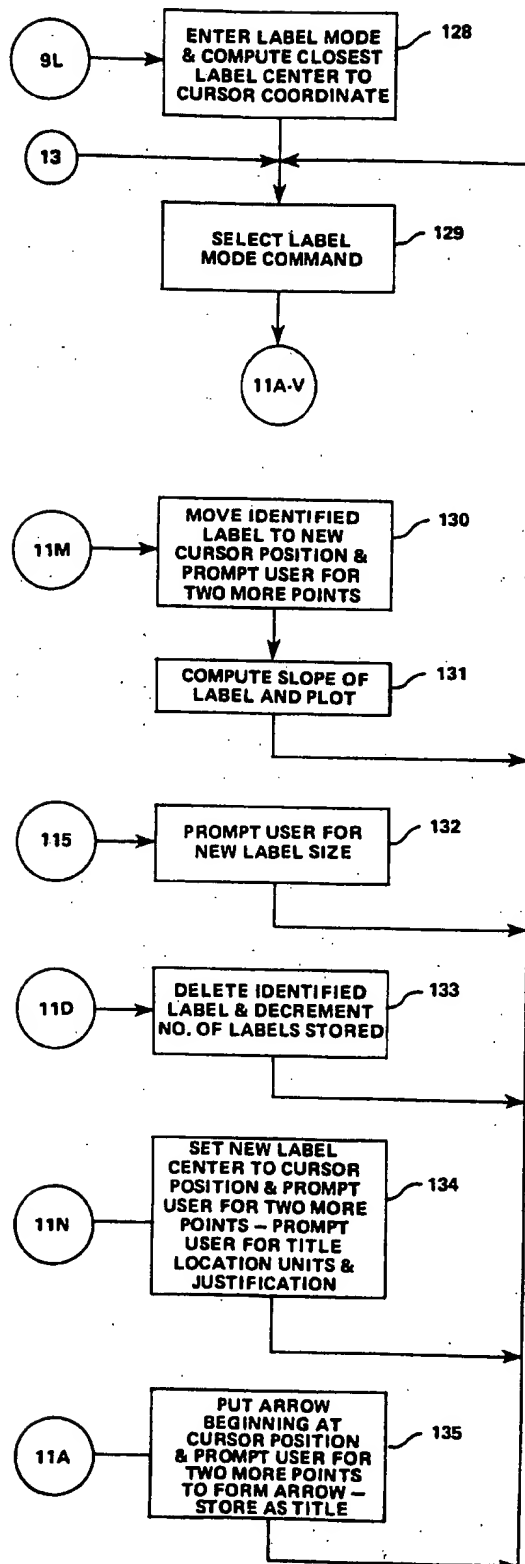


FIG. 3H

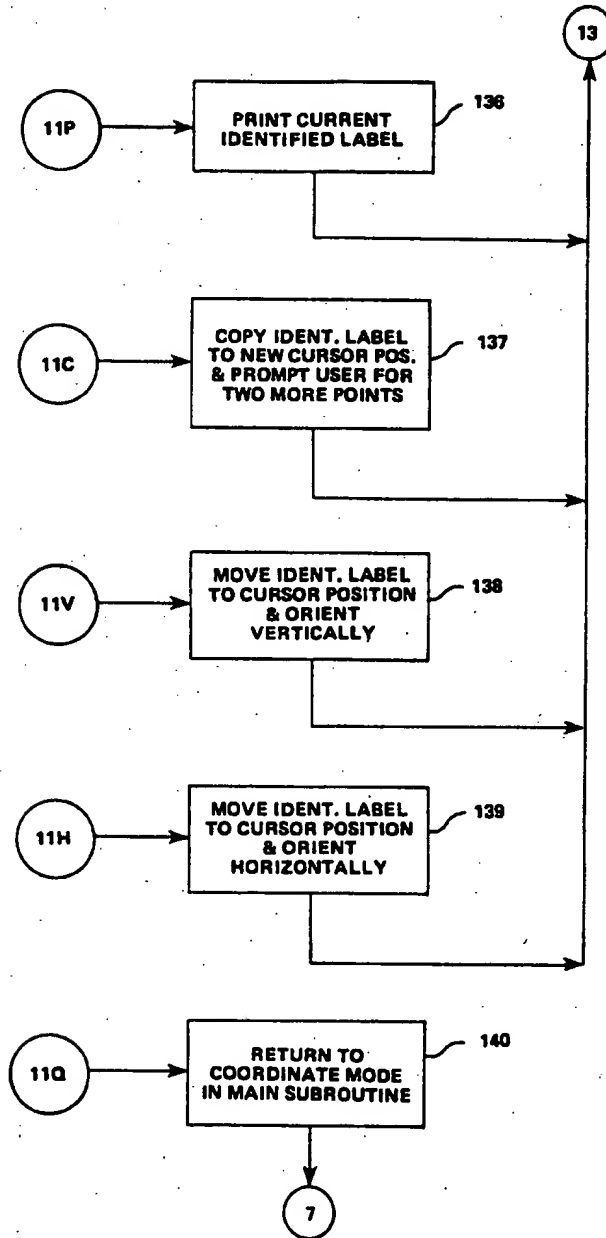


FIG. 3I

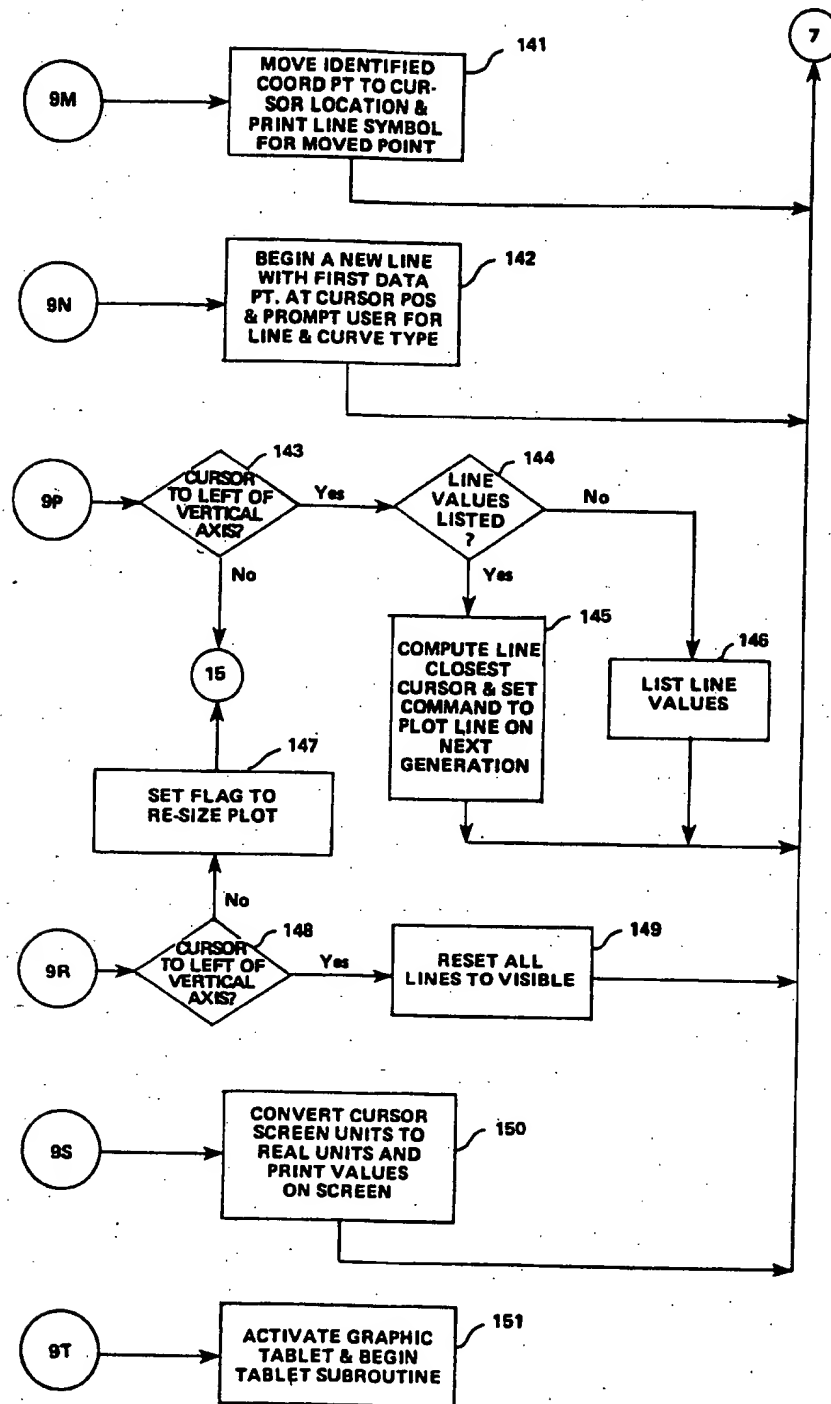


FIG. 3J

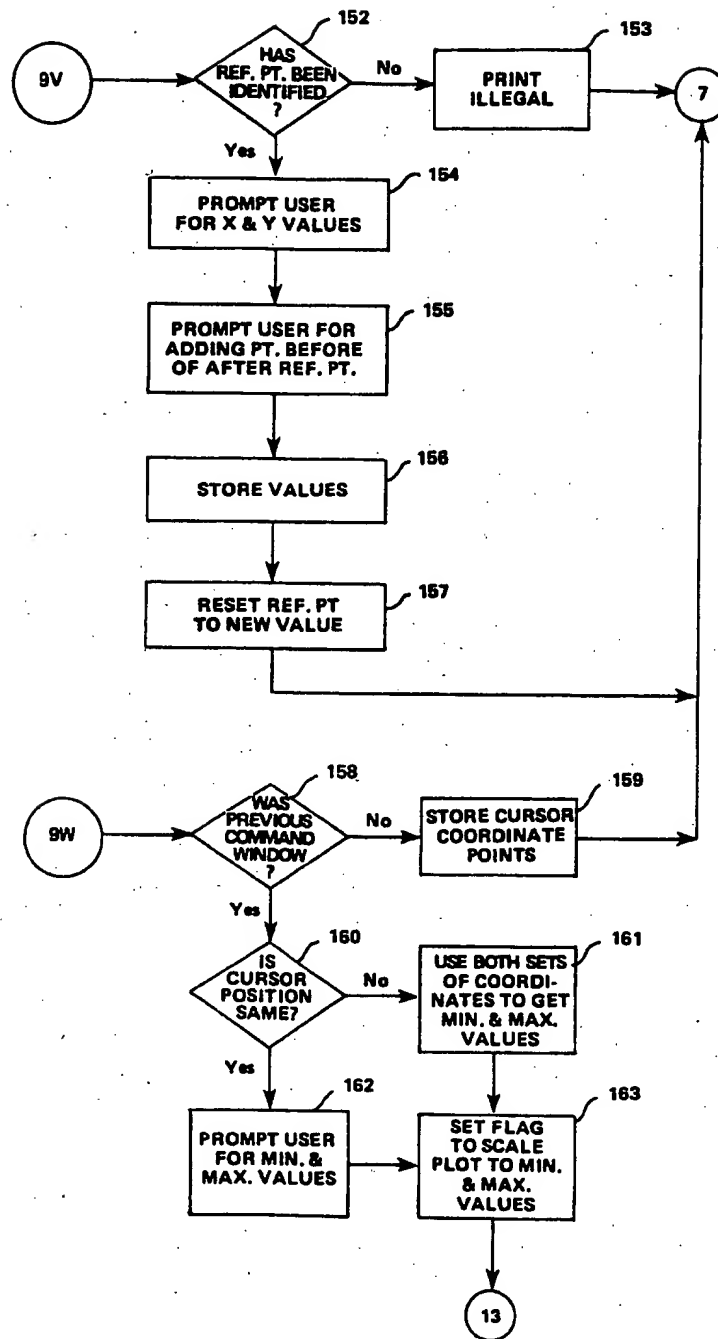


FIG. 3K

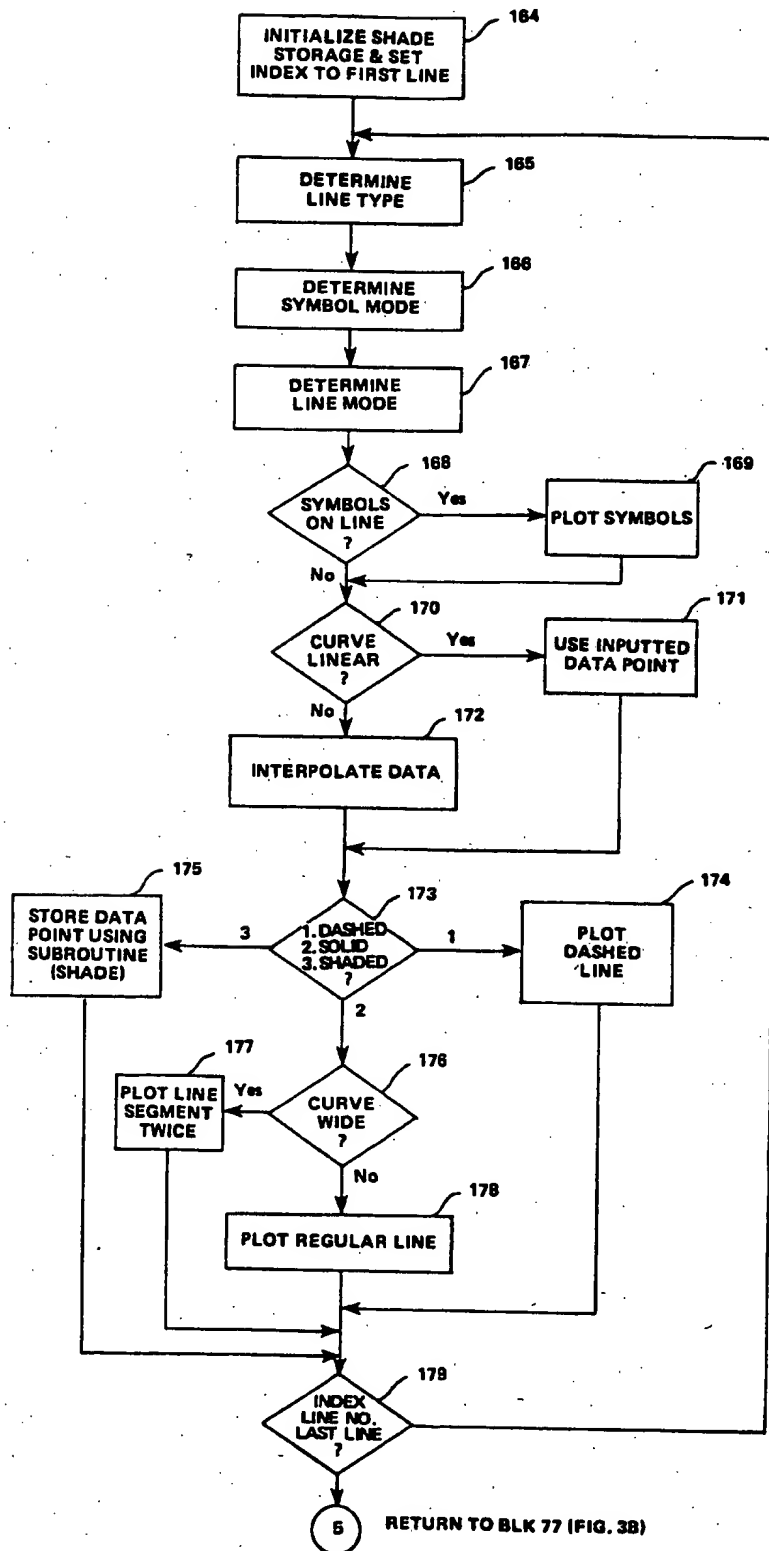


FIG. 3L

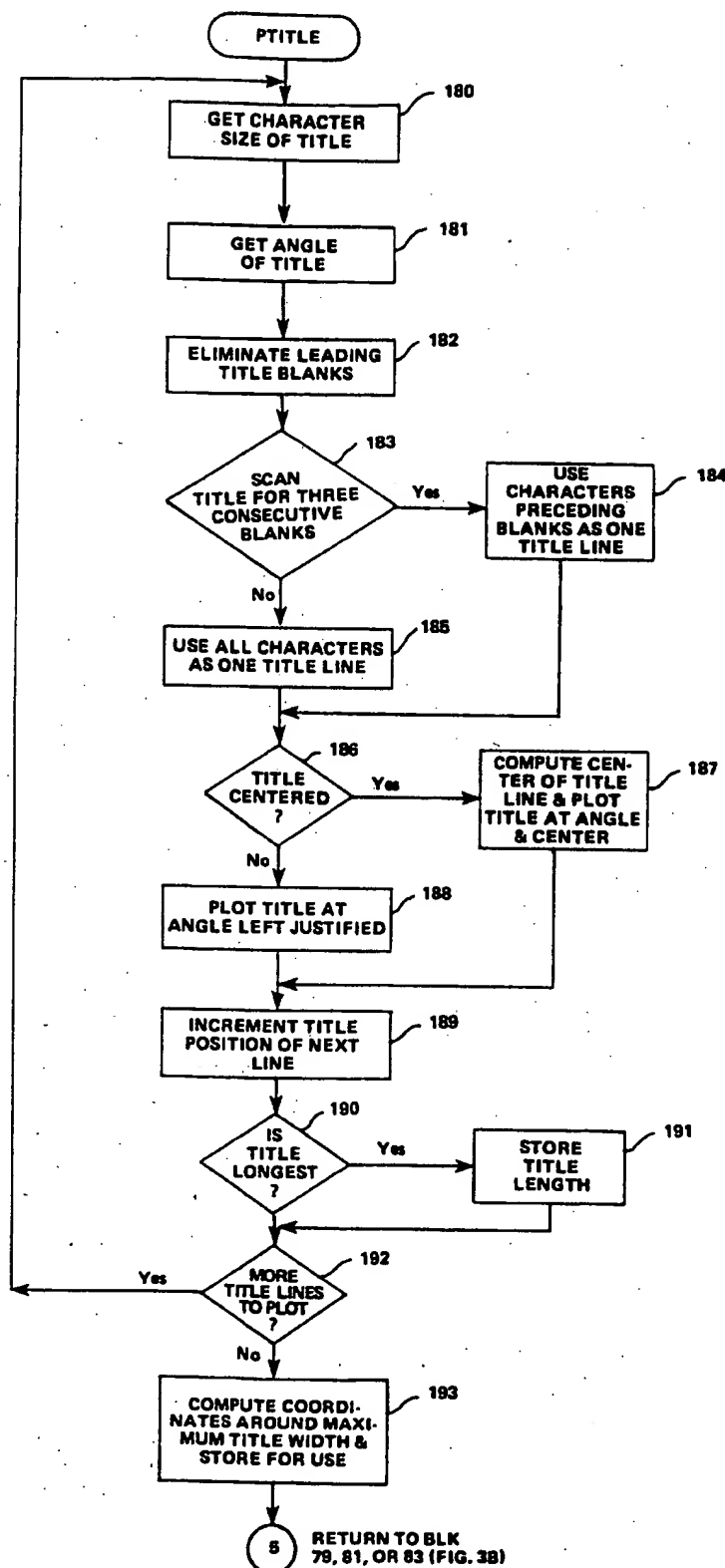


FIG. 3M



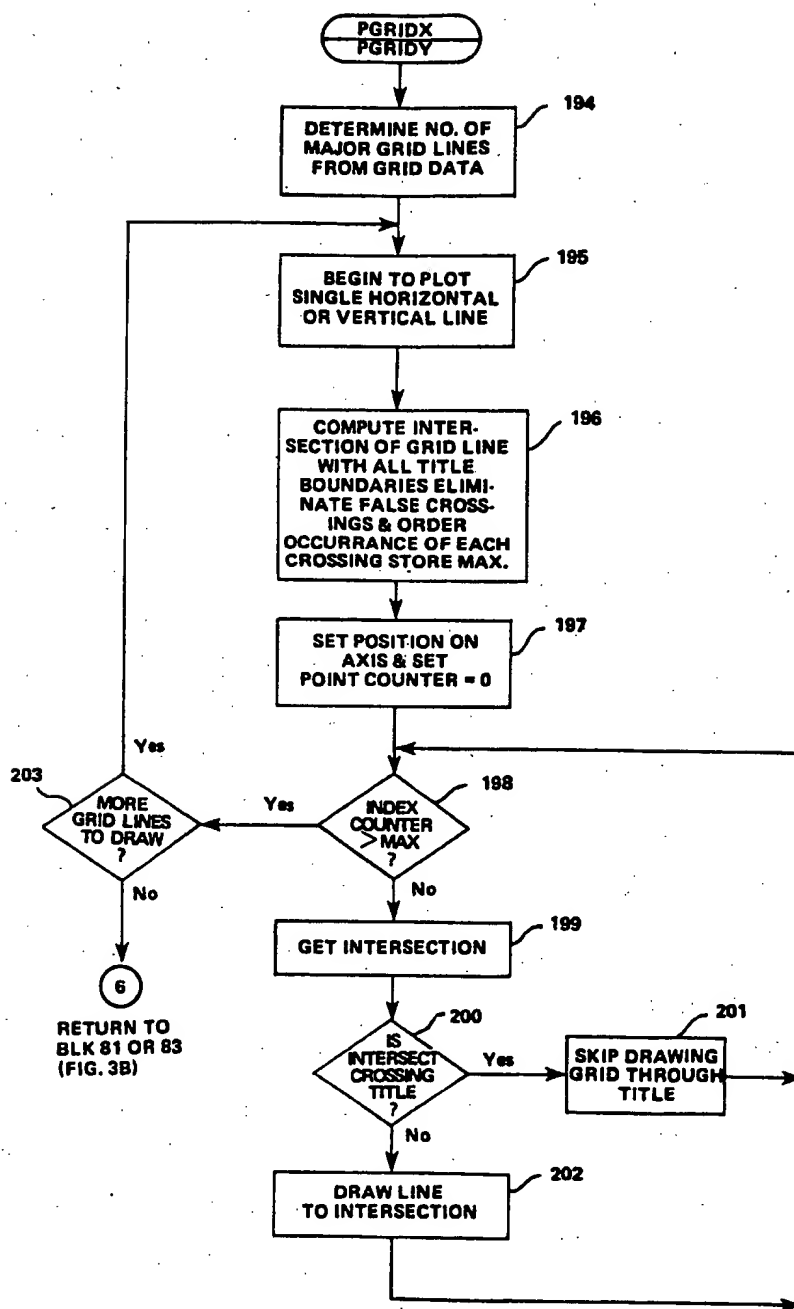


FIG. 3N

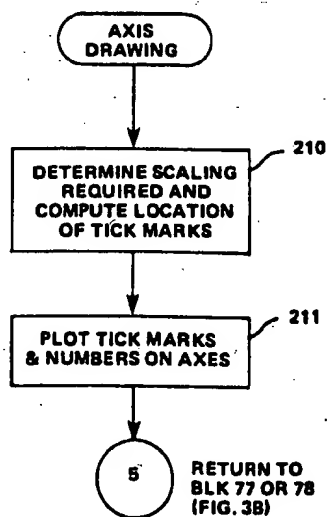
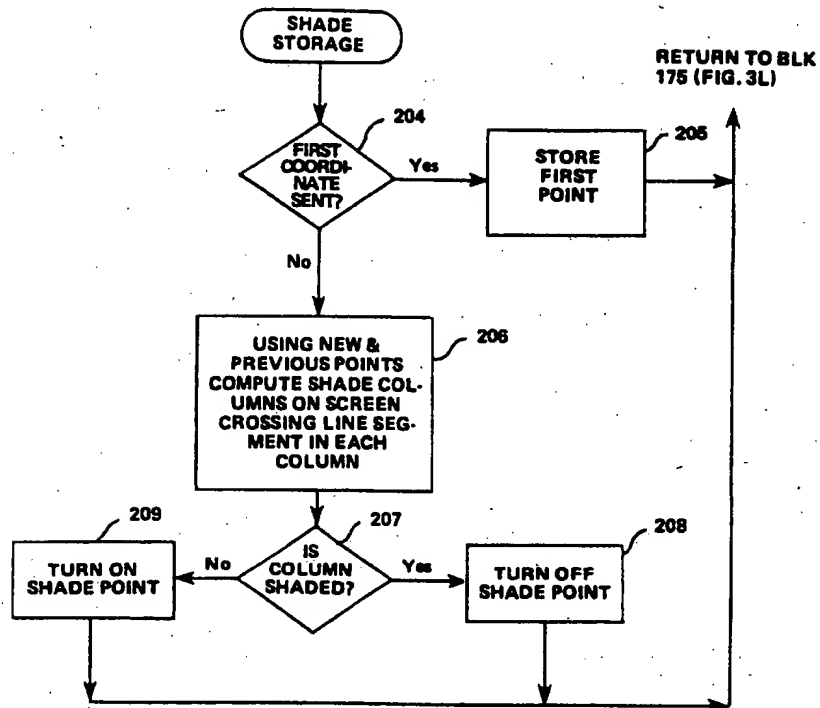


FIG. 30

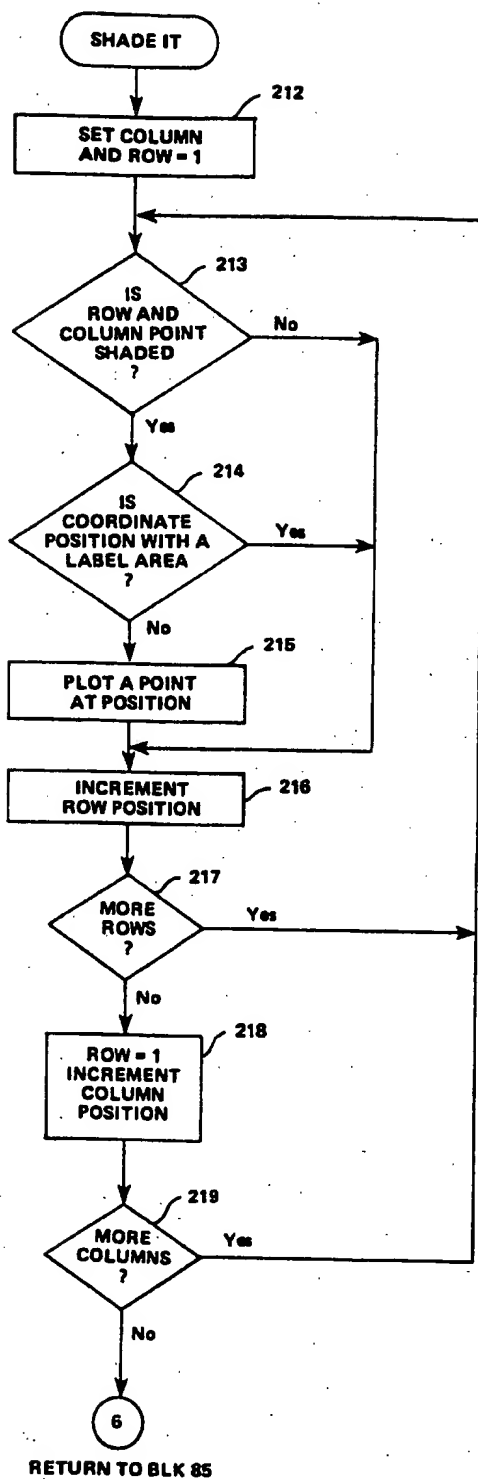


FIG. 3P

# COMPUTER-AIDED PROCESS FOR GENERATING CAMERA-READY GRAPHICAL ARTWORK

## STATEMENT OF GOVERNMENT INTEREST

The invention described herein may be manufactured and used by or for the Government of the United States of America for governmental purposes without the payment of any royalties thereon or therefor.

## APPENDIX

An appendix consisting of 51 pages is included in this application.

## BACKGROUND OF THE INVENTION

The present invention relates to graphic arts in general and more particularly to an improved process for automatically generating camera-ready graphical artwork with the aid of a computer.

Graphical artwork, more specifically, graphical plots are commonly used as a visual aid to display a substantial amount of information regarding the coordinate relationships of certain variable physical quantities. In addition to the plotting of the basic coordinate data, typically for selected values of a variable factor or condition, such graphical plots generally include a variety of reference lines or curves as well as shading patterns for ready observation and interpretation of the data. Large quantities of these highly informative graphical plots, often found in scientific works and technical reports and manuals, are usually printed using conventional methods of photolithography that require production of a high-quality reproduction copy of the graphical plots in intricate detail, ready for photographing by a process camera.

Commonly known as being camera-ready, such high-quality reproduction copies of the graphical plots have been difficult and time-consuming to produce as well as to edit and correct if necessary. Hand-drawing and editing of the plots by skilled draftsmen, although satisfactory from a quality standpoint, continues to be painstaking and costly. Machine-drawn plots can be produced in substantially less time and have generally been adequate in quality and detail. However, such machine-drawn artwork still requires manual "cut-outs" and "paste-ons" to meet camera-ready requirements.

## SUMMARY OF THE INVENTION

Accordingly, it is a main purpose and general object of the present invention to provide an improved process implemented by a computer for generating graphical artwork of a finished quality ready for photolithographic reproduction.

It is a more particular object of the present invention to provide a computer-aided process for producing original camera-ready graphical plots in full detail without requiring any manual drafting labor.

It is a further object of the present invention to provide an automated process for creating revised camera-ready graphical plots that permits custom editing and correcting of existing plots quickly and precisely without manually redrawing revisions and affixing those revisions to the existing plots.

It is a still further object of the present invention to provide a computer-aided process for graphical artwork generation that is cost effective, reliable in perfor-

mance, and easily adapted to existing automated graphic art equipment.

Briefly, these and other aspects of the present invention are accomplished by a computer-aided process for automatically generating a camera-ready hardcopy of a graphical plot upon command instructions inputted via a conventional storage tube graphics display terminal having an addressable cross hair cursor and a keyboard. In accordance with an interactive graphics code or program, tabular data coordinates stored in computer file form are retrieved and plotted on appropriately titled and scaled axes with the plotted coordinates being interconnected along curves formed of a smooth or linear nature by interpolation. The graphical plot viewed on the display terminal is further enhanced by inclusion of labels, shaded areas, and reference symbols and characters prior to printing out the hardcopy on an associated hardcopy unit coupled to the display terminal.

For a better understanding of these and other aspects of the present invention, reference may be made to the following detailed description taken in conjunction with the accompanying drawings in which like reference numerals designate like parts throughout the figures thereof.

## BRIEF DESCRIPTION OF THE DRAWING

FIG. 1 is a general block diagram showing the hardware used to implement the process for automatically generating camera-ready graphical plots in accordance with the present invention;

FIG. 2 is a general flow diagram of the computer-aided process of the present invention, showing the broad stages of data processing steps and their standard sequence; and

FIGS. 3A-3P, inclusive, represent a more detailed flow diagram of the computer-aided process in accordance with the present invention.

## DESCRIPTION OF THE PREFERRED EMBODIMENT

Referring now to FIG. 1, the process for automated graphics generation according to the present invention is primarily aided and implemented by a general purpose digital computer system 10 serving as the host and programmed to operate in accordance with an interactive graphics code, described in greater detail hereinafter.

It should be noted that the interactive graphics code, used by the host computer system 10 to interpret and perform the interactive requests of the user is also described in the Appendix, which is the source code listing of the interactive graphics code written in FORTRAN IV. This source code listing of the instructions, routines, and other contents of the interactive graphics code in appropriate sequence may be implemented, for example, on a CDC Cyber Model 175 computer using a FORTRAN computer CDC Version 438, a linker/loader CDC Version 1.5538 for Network Operating System (NOS) 1.4, and a library module TEKTRONIX Plot 10.

Stored with sets of tabular data coordinates in computer file form, digital computer system 10 is coupled to a conventional storage tube graphics display terminal 12 which accepts data from and sends data to the computer system. Display terminal 12 is equipped with a screen 14 on which output from computer system 10 is displayed to a user, the screen being the face of a storage tube

device (not shown), typically a cathode ray tube (CRT), which maintains a display once written, for an indefinite period of time until an erasure is made. Display terminal 12 is further provided with a control panel 16 on which a keyboard 18 is located for allowing the user to enter alphanumeric (A/N) data onto the screen 14 and onto the computer system 10. A cursor control 22 also located on the control panel 16 is employed by the user to address a graphic cross-hair cursor 20 movable across screen 14 to specify positional input of data. The display terminal 12 with the aforescribed features is a commercially available unit, one suitable such unit being the Tektronix Model 4015 with its associated graphics software. A conventional hardcopy unit 24 compatible with display terminal 12 is coupled thereto for producing, in accordance with the present invention, a camera-ready, hardcopy 26 of a graphical plot produced on the screen 14.

Referring now to FIG. 2, Block 31 represents a general start up procedure in which the user establishes communication with the host computer system 10 and commands use of old data files and eventually begins execution of the interactive graphics code. Block 31 is a query as to creating a new data file or using the old one. If the user's response is new data, then a query for axis labels and other minimum graphical data is indicated. In Block 31, the old data is automatically inputted into storage along with labels and other information. Block 33 is the top of the interactive loop in which the user physically moves the coordinates on screen 14 via cross-hair cursor 20 and keys a single character on keyboard 18. This single character and the coordinates comprise a user command. Block 33 additionally performs other data checks such as, if this is a new plot (no data to plot), a create line command is automatically executed. Block 34 checks for a valid command. An invalid command sends control back to block 33. Blocks 35 thru 38 interpret the command, use the coordinate data, if required, and act appropriately on the plot.

Three groups of commands are available: label, coordinate and plot. Block 36 only concerns labeling. Block 37 concerns only line forming, coordinate points and shading. Block 38 concerns the general plot attributes, such as size, grid and line types. Block 39 thru 46, inclusive, follow a command to plot the data. Block 39 uses the minimum and maximum data ranges and the grid requirements, and picks an appropriate scale on each axis, finally plotting the axis and labels. Block 40, besides plotting other labels, saves the four coordinate points around the label for later use. Blocks 41 thru 44, inclusive, are concerned with lines on the plot. Options are available for smooth curve, linear, or just symbols. In addition, lines can be one of 9 dash line types with options for solid thick or thin lines, as well as closed or open curves. Block 45 plots the grid as specified and uses the label coordinate point data generated in block 40 such that no grid line extends thru a label. Block 46 completes the plot by determining shading regions and plotting them with a uniform grid of dots again avoiding shading of labels. The end result of block 46 is that the user has on the screen 14 the latest edited plot. At this point control is transferred to block 33 for more editing or termination of the plot.

Referring now to FIG. 3A, in Block 50 the user turns on the display terminal 12, connects to the host computer system 10 and begins execution of the interactive graphics code. Block 51 prompts the user as to product and date of version of the graphics code. In addition, a

query is issued for a communication transmission rate with the host computer system 10. In block 52, the user picks an appropriate code number 1-4, indicating the desired rate of character transmission or baud rate. For example, codes 1-4 typically designate baud rates of 1200, 2400, 4800 and 9600 characters per second, respectively. Block 53 checks the baud rate code and passes to block 54 if valid; otherwise, the process moves back to block 52. Block 54 makes a read attempt to determine if the user has connected through the host computer system 10 another file containing an old plot data set. Block 55 checks this read and if an end of file was found, then a new plot is implied and the user is prompted for minimum data (plot axis labels for storage) in block 57. Block 58 queries the user for decimal places for each data group. These data are those generated during the plot creation and used to generate a data file of the created plot for storage in the files of the host computer system 10. Blocks 59 and 60 check and query the user for these data. Referring back to the other branch of block 55, in block 56 old plot data is read from file storage and readied for use. Block 61 sets the communications transmission rate and the character size on the screen 14 of display terminal 12 to small. Block 62 begins the interactive plotting in subroutine TIGPPR. Block 63 sets the grid and range of data to either user input values or default. The default data range is the minimum and maximum values of the X and Y data, respectively. Block 64 initializes a set of points to the serial coordinate data forming a linked list. The linked list is used to efficiently add or delete a coordinate data point to a line or shaded area. Block 65 sets the graphics software associated with the display terminal 12 for use of stroke generated characters rather than a composite (one of four fixed size horizontal characters). This is required in order to generate labels at any size and rotation. Block 66 operates on the linked list, determines the number of lines and where each line begins in the linked list data storage.

Referring now to FIG. 3B, block 67 checks the number or coordinate points and, if it is zero (a new plot), the user is prompted and sent to the command mode. If data exists, then a check is made in block 69 to see if this data has been plotted once. If not, block 70 prints the main plot label and GRID/DATA range values. A window on screen 14 (physical plot size) is now set in block 17. Block 72 checks the validity of data ranges. If they are not valid, block 73 computes minimum and maximum values, respectively. Block 74 determines data scaling parameters from minimum and maximum X and Y values. The scales being preset to multiples of 1, 2 or 5. Block 75 computes the scaling of real units to the screen units, typically  $4096 \times 4096$ , on the display terminal 12. Block 76 checks for first time plot and gives the user a chance to change size, grid and other parameters. Blocks 77, 78 and 79 are reference to respective subroutines for drawing curves and labels, shown in greater detail in FIGS. 3L, 3M and 3O, respectively. Blocks 80 thru 83 determine if default axis labels are drawn, and draw all other labels using the subroutine PTITE, shown in greater detail in FIG. 3M.

Referring now to FIG. 3C, block 84 checks for existence of shading. If shading exists, then subroutine SHADIT, shown in FIG. 3P, is called in block 85. Similarly, both X and Y Grid requirements are checked and subroutines PGRIDX and PGRIDY shown in FIG. 3N are called as needed. Block 90 turns off the

stroke characters since they are not needed in the command mode.

Referring now to FIG. 3D, in block 91, the user physically positions the movable cross-hair cursor 20 to a position on screen 14 after which a single character on keyboard 18 is depressed. The character depressed is used to determine the coordinate command and the position of the cross-hair cursor 20 on screen 14 used to compute a coordinate position X and Y. Block 92 checks for a valid command by performing a table look-up using the ASCII code for the character keyed. The ASCII is the American Standard Code for Information Interchange, a standard code consisting of 7-bit elements for information interchange among data processing communications systems. Block 92 flags the character as invalid, i.e. illegal, and re-prompts user in block 91. A valid coordinate command from the user continues the process as desired, and such valid commands are as follows:

VALID COORDINATE COMMAND SUMMARY	
COMMAND	DESCRIPTION
A	add point after reference point
B	add point before reference point
C	identify closest data point (to cross hairs)
D	delete data point
E	exit
F	format (change line type)
G	input grid data
H	halt graphics tablet mode
K	kill line with closest data point to cross hairs
L	ENTER label mode and locate closest label to cross hairs
M	MOVE identified data point to new position
N	input a new data point beginning a new line
P	re-plot the current data using same window scale
Q	quit label mode
R	repeat the current data (but resize it to fill screen)
S	show real values X and Y at current screen position
T	Enter graphic tablet mode
V	input direct coordinate values for X and Y
W	Window data P replot to new minimum and maximum values

The above coordinate commands, prefixed each respectively by the number "9", are shown in FIGS. 3E, 3F, 3G, 3H, 3J and 3K and are described in greater detail hereinafter with appropriate reference to those figures.

Referring now to FIG. 3E, block 94 checks to see if this is the first point of a new plot; if it is, then control is transferred to the appropriate coordinate command for inputting a new line. The command "A" (99A) basically adds a point in storage after a reference point, the reference point being set by the command "C" (9C), shown in FIG. 3F. Block 95 checks to see if the user had set a reference point, a transfer to block 93 indicating a point not set and an invalid command. Block 96 checks to see if the point is within the plot frame on the screen 14 of display terminal 12. Block 90 plots a character indicating the line type at a coordinate position selected by the user. Block 98, using the linked list storage, adds the new point into the list after the reference point, and changes the reference point to the new point just added. Coordinate command "B" (9B), also shown in FIG. 3E, follows the same logic as the "A" command except as to placement of the new point relative to the reference point. Via block 103, the "B" command puts the point before the reference point in storage. The linked list is used the same way except the coordinate values are

swapped so that the link list remains the same as in the A command mode.

Referring now to FIG. 3F, the "C" command (9C) establishes a reference point by finding the closes coordinate point per block 104. Blocks 105 and 106 check the point and plot a line character symbol if the point is within the plot frame. Block 107 moves the linked list forward pointer to the next set of coordinate points and stores the positions and line number of the reference point. The "D" coordinate command (9D), also shown in FIG. 3F, deletes a coordinate point and performs the same operations in blocks 108, 190 and 110 as the respective blocks 104, 105 and 106 for "C" command. The "D" command differs, however, from the "C" command in that in Block 111 the point is deleted from the linked list. This is accomplished by moving the forward coordinate set, as pointed to by the forward pointer, to the deleted point position. The unused data storage space is then added to another linked list which is a list of unused storage. Block 112 turns off the reference pointer flag since the point was just deleted. The "E" or exit coordinate command (9E) is executed per block 113 which terminates the plot, returns to the main program, and converts the linked data to serial data and outputs the data with all corrections, labels and arrows to the file of host computer 10.

Referring now to FIG. 3G, the "F" or format command (9F) can change the line characteristics in terms of linear, smooth, symbols only, no symbols, closed curve, or shade. The user positions the cross-hair cursor 20 either to the left or right of the vertical axis of the plot. Block 121 checks for this relative position of cursor 20. If the position is to the left, block 122 is used and the line type and curve type are changed for all lines of the entire plot. If the cursor position is to the right of the vertical axis, block 123 is used and the format change only applies to the indicated line. The lines and associated symbols are listed in a column at the left top of the terminal display screen 14. In block 115, the vertical coordinate position is used to determine the closest line and symbol. The line number is derived from this operation. Blocks 116, 117, 118 and 119 check that value and keep it within the range of one to the maximum number of lines plotted. Block 120 prompts the user for the new line type value and new curve type value. Blocks 121, 122 and 123, as previously discussed, determine how to apply new line type and curve type.

The "G" command (9G), also shown in FIG. 3G is executed via block 124 and is used to change the grid format. The user selects the number of major and minor tick marks and the frequency of grid lines on both axes. Via block 125 the "H" Command (9H) is effected. The "H" command is used to terminate the tablet command mode, the tablet mode being analogous to the screen mode in the context of the command and coordinate entry. The "K" or kill command (9K) is executed via block 126 and 127 and is similar to the "D" or delete command except that the "K" command deletes the entire line.

Referring now to FIG. 3H, the "L" command (9L) enters the label mode via block 128. The label mode keys a new set of single key commands which only affect labeling of the plot. The initial "L" command enters the label mode and uses the coordinate position to compute the closest label. The following is a summary of label mode commands:

LABEL MODE COMMAND SUMMARY	
COMMAND	DESCRIPTION
L	Identify closest label and plot point showing its reference point
M	Move identified label to indicated cursor position
S	Change label size
D	Delete label
Q	Quit label mode return to coordinate mode
N	Add new label at current cursor position
A	Add arrow to plot
P	Print current identified label
C	Copy identified label to new cursor position
V	Move label to new vertical position
H	Move label to new horizontal position

These label mode commands, like the coordinate mode commands, are prefixed by the number "11" for designation in the drawing figures and are described herein below.

In accordance with the "M" label mode command (11M), the user moves an identified label to a described position using cursor 20. The user is then prompted for two more coordinate points. These points are used to compute the slope of the label in block 131. The label is then printed at the indicated position along the new slope. Block 132 represents the "S" label mode command (11S) used to change the size of a label. Block 132 also checks to insure that a label has been identified using the initial "L" command. Block 133 executes the "D" label mode command (11D), deleting the previously identified label. The "N" label mode command (11N) executed via block 134 inputs a new label and its attributes, i.e., justification and storage. Block 134 also prompts the user for 2 additional points required for slope computation. In block 135, the "A" label mode command (11A) is implemented. The user positions the cursor 20 at the arrow tail (non-pointed end) and keys "A". Two additional points are requested giving a broken arrow with the final point as the arrow head.

Referring now to FIG. 3I, block 136 affects a "P" or print label command (11P). In accordance with block 136, the current located label is re-printed. This provides a means for the user to identify which label is located when the initial "L" command is used in clustered labels. Block 137 implements the "C" label mode command (11C), generating a copy of the identified label and promptly the user for two additional points to define the slope of the new copied label. The "V" label mode command (11V) implemented via block 138 is a quick label move to the indicated position, with the orientation vertical, reading from the bottom to the top on right side. Similarly, block 139 executes the "H" label mode command (11H), performing an analogous label move, with the label set horizontal. Both the "V" and "H" label mode commands in blocks 138 and 139 are fast versions of the previously described "M" label mode command (11M). Block 140 represents the "Q" label mode command (11Q) that exits the label mode and returns the user to the coordinate command mode.

Referring now to FIG. 3J, the remainder of the coordinate commands are shown. In block 141, the "M" coordinate command (9M) is executed, moving the previously identified coordinate point to a position as indicated by cursor 20. Block 141 also checks to insure that a previous point has been identified as the reference point via the "C" coordinate command and block 104. Block 142 prompts the user for a new line value as well as line and curve type in accordance with the "N" coordinate command (9N).

The coordinate point at the cursor 20 is identified as the first point of the new curve. In addition the reference point is automatically set to the new point.

The "P" plot coordinate command (9P) is executed in a series of steps represented by blocks 143-146. The first step of the "P" coordinate command, taken in block 143, is to determine the vertical position of the cross-hair cursor 20. If the position is to the right of the vertical axis, then control is transferred to block 65 to FIG. 3A, causing the graphics code to replot the data. If the cursor 20 is to the left of the vertical axis then further checks are performed. In block 144, a check is made to see if the line values and symbols have been already listed, i.e., in column format, left of vertical axis. Block 147 lists these line values and symbols if they have not been listed and in block 145 the line value of the closest line, listed in block 146, to the cursor 20 is determined. This closest line is toggled to plot in the next "P" command, and gives the user the capability to edit, plot and correct several lines overlapping on each other by plotting each one separately. The "P" command based on position of cursor 20 left of the vertical axis, can be thought of as a pick to live to plot command. In block 148 the "R" or restore/reset coordinate command (9R) is initiated. As with the "P" command and block 143, the vertical coordinate position of cursor 20 is used to determine the effect of the command. To the left of the vertical axis, the "R" command restores all lines to plot visibility. To the right, the "R" command resizes the plot to fit the minimum and maximum values contained within the coordinate data. In block 150, the "S" or "show" coordinate command is executed, the coordinate values being used to compute the actual real units at the cross-hair location of cursor 20 and these values being displayed on the screen 14. In block 151, the control is transferred, in accordance with the "T" coordinate command (9T), to the graphics tablet using a "mouse" to position and key coordinate commands.

In FIG. 3K, block 152 initiates the "V" coordinate command (9V). The "V" command is similar to the "A" or add point command except that the user directly inputs the exact values. Block 152 checks for a reference point and block 154 prompts the user for one set of coordinate values. Block 155 prompts the user to store before or after the reference point. Block 157 resets the new point to the reference point and transfers to the next command. Block 158 initiates the "W" or window coordinate command. In this command, the user is prompted for two sets of coordinate points from which a rectangular window is formed and the plot size enlarged to fill screen 14 with data in that window. Block 158 gets the first set of coordinate points and block 160 checks for closeness of the two sets of coordinate points. Block 161 uses the two sets and redraws the plot enlarged on that window. Block 162 is transferred to when the user has not moved the cursor 20 for the second set of coordinate points, implying that direct input of window coordinate points is being requested. Block 162 performs this request. Block 163 computes minimum and maximum values and transfers to replot the data with the newly specified minimum and maximum.

Referring now to FIG. 3L, the DRAWIT subroutine is shown. In block 164 the shading storage, a bit map representation of the entire screen 14 is set to zero i.e., no shade. Block 165 determines the line type, curve or

linear, and block 166 plots the symbols, if required. Block 167 determines the line mode, solid or dashed. Block 168 begins to plot the curve, determining if symbols at each coordinate point are required, and block 169 instructs the plotting of the symbols. Via block 170, if the curve type is linear, then a straight line is drawn between data points. Data interpolation using the Hiroshi Akima technique is executed via block 172, using all the points on the given line and generating short line segments. Block 173 determines the type of line, i.e., dashed, solid or shaded. Block 174 plots the line as dashed using one of several different pre-programmed dash sequences. Block 175 imposes the shading on the plot using the following algorithm: in block 164, a bit map representation of the plot is initialized (about 20,000 points per plot); as the curve in which shading was requested is drawn, the points in the columns beneath the curve are examined; if a point (single bit) is not already shaded (bit is zero), then it is set to shade (bit set to 1); conversely if the bit is set to shade (bit equals 1), then it is reset to no shade (bit equals 0). This algorithm permits any shape defined by a curve, open or closed to be shaded. Further related discussion is found hereinafter regarding the subroutine SHADE. Block 176 determines if the wide solid line is required and block 177 plots the wide curve by performing a "hem stitch" motion as the curve is drawn. Block 178 simply draws the regular line, single width. Block 179 checks to see if any more lines are to be drawn, and if not returns control to block 77 in FIG. 3B.

Referring now to FIG. 3M, the PTITLE subroutine (plot titles) is shown. Block 180 gets the size of the title as inputted or changed by the user. Block 181 uses the two slope points also inputted by the user to compute the title angle. Block 183 eliminates all leading blanks from the label. Block 183 begins scanning the title for three consecutive blanks used as a signal to terminate this line and begin a new line. Block 184 does the termination. Likewise, if the scan does not find three consecutive blanks, then the entire title is assumed on one line. Block 186 checks user input to see if title is centered or left shifted. Block 189 plots the title and increment to the next line. Block 190 checks the length of the line and stores the maximum line length. Block 192 checks for more title lines in the same title and transfers when finished to block 193. Block 193 stores the maximum line length and computes four coordinates around the title and stores these for later use. FIG. 3N shows the flow for the grid drawing subroutine PGRIDY/P-GRIDX. Block 194 determines which grid lines are to be drawn. Block 195 moves into position to begin the grid line on either axis. Block 196 computes all possible intersections of this line with all title boundaries as determined in the PTITLE routine of FIG. 3M. False crossings are eliminated and the sequence of real crossings is stored in the order of crossing. Block 197 initializes the position and counter while block 198 determines if the end of the current grid line is reached. Block 203 checks for more grid lines if the current end is reached. Block 199 computes the next intersection of the grid line with the next title boundary it crosses and block 200 checks to see if drawing from the current

position to intersection is crossing a title block. If it is, then block 201 skips drawing and moves to the computed intersection. Block 202 is used to draw a line to the intersection when a line is not passing thru title. This process is repeated looping back thru to block 198 until the end of the current grid line is reached.

Referring now to FIG. 3O, the SHADE subroutine is described. Block 204 checks for the first point of a line segment and block 205 stores it. In succeeding passes thru SHADE control is transferred to block 206. Block 206 takes two points and examines all columns of points between the X-axis and the line spanning the interval of the points. Within each column beneath this interval each possible shade point is checked. If the point is already shaded, it is unshaded and similarly, a point unshaded is shaded. This process is repeated until the line is traversed to its end. Also in FIG. 3O, the AXIS subroutine is shown. Block 211 uses the minimum and maximum values determined from the coordinate data and generates scaling in a multiple of 1, 2 or 5 units. Tick marks and numbers required on axis are computed and plotted.

Referring now to FIG. 3P, subroutine SHADEIT is shown. SHADEIT actually plots the points assembled with the SHADE subroutine shown in FIG. 3O. The exception is that points within a title block are omitted from shading. Block 212 sets the column row position and begins checking all 20,000 possible points. Block 213 checks point for shade. If point is shaded block 214 checks to see if shaded point is within label. If not, block 215 plots the point and block 216 increments the row counter. Block 217 checks for more rows and block 218 resets to Row 1 and moves to the next column. Block 219 then checks for more columns, and if none are found, return to block 85.

Therefore, it is apparent that the present invention provides an improved process implemented by a computer for generating graphical artwork of a finished quality ready for photolithographic reproduction. More particularly, the present invention provides a computer-aided process for producing original camera-ready plots in full detail without requiring any manual drafting. Furthermore, the disclosed invention provides an automated process for creating revised camera-ready graphical plots that permits custom editing and correcting of existing plots quickly and precisely without manually redrawing revisions and affixing them to existing plots. In addition, the disclosed computer-aided process for graphical artwork generation is cost effective, reliable in performance, easily adapted to existing automated graphic art equipment.

Obviously, other embodiments and modifications of the present invention will readily come to those of ordinary skill in the art having the benefit of the teachings presented in the foregoing description and drawings. It is therefore to be understood that various changes in the details, materials, steps, and arrangement of parts, which have been described and illustrated to explain the nature of the invention, may be made by those skilled in the art within the principle and scope of the invention as expressed in the appended claims.



APPENDIX  
FORTRAN LISTING

CUTIGS

TIGS000

```

C*** USE P4662 PROC FILE
      PROGRAM UTIGS(INPUT=101,OUTPUT,TAPE1=101,TAPE7=101,TAPE5=INPUT)
C *****
C****
C****      ULTRA TIGS TPLLOT INTERACTIVE GRAPHICS SYSTEM
C****
C****      E CADDY REVISED SEP 82
      COMMON/IITLE/IIT(50),JTT(50),KTT(50),LIT(50),MIT(50),STT(50),
1 XTT(50),YTT(50),XUU(50),YUU(50),XVV(50),YVV(50),ITC(8,50),NTITE
2 ,AXMIN,AYMIN,AXMAX,AYMAX,IXDEX,IYDEX,IGRIDX,IGRIDY,ITMX,ITMY,
3 ICHAR,IWIN
      DIMENSION LT(7),XV(50),NPTS(50),X(1000),Y(1000),Z(50),A(200)
      DIMENSION IRATE(4)
      DATA NPLCT/0/
      DATA NPTS,XV/50*0.50*0./
      DATA IEND/10HEOT /
      DATA IRATE/120,240,480,960/
10 FORMAT(A5,7A10)
      REWIND 1
      REWIND 7
      PRINT *,"UTIGS VER 1.0 02-08-83      DEFAULT IS 9600 BAUD"
      PRINT *,"AND MAXIMUM DECIMALS ON TAPE7 OUTPUT"
      PRINT *,"ENTER C R> TO CHANGE OR  R> TO CONTINUE"
      IER=960
      IDFLT=1
      READ 30,IBZ
      IF(EOF(5).NE.0.)GO TO 29
      IDFLT=0
      PRINT 21
21 FORMAT(* ENTER BAUD RATE CODE *,/,
1 * 1=1200, 2=2400, 3=4800, 4=9600*)
      CALL GETIN(1,Z(1))
      IER=Z
      IER=MAXO(1,MINO(IER,4))
      IER=IRATE(IER)
C****
C****      FILE IS NOT BEING CREATED READ IT FROM TAPE1
C****
29 IC=0
30 FORMAT(1R1)
31 READ (1,10) LNO,LT
      IF(EOF(1).EQ.0.)GO TO 32
      IC=1
      PRINT *,"CREATION MODE"
      GO TO 40
32 IF(LNO.EQ.10H) ) GO TO 251
C****
C****      CALL IN 2 VALUES
C****
      CALL TABR(LZ,L,MT,KZ,2,1)
C****
C****      READ IN Y,X,FX,YZ DATA FOR NON CREATION RUN
C****
      CALL TABR(LY,L,MT,KY,A,1)
      CALL TABR(LX,L,MT,A,X,1)
C*** SAVE STORAGE LOCATION OF LAST X VALUE
      JXS=0

```

TIGS0003

TIGS0004

TIGS0005

TIGS0007

TIGS0010

TIGS0015

TIGS0014

TIGS0015

TIGS0016

TIGS0017

TIGS0024

TIGS0025

TIGS0026

TIGS0027

TIGS0028

TIGS0029

TIGS0030

TIGS0032

TIGS0033

TIGS0034

```

      CALL TAER(LF,L,NT,N,Y,1)
      NTT(1)=NT
      LTT(1)=L
      GO TC 100
40 PRINT 50
50 FORMAT(* ENTER TABLE TITLE CARD*/
1 * (COLUMNS 1-5 SHOULD BE THE TABLE REFERENCE NUMBER)*)
      READ 10,LNC,LT
C****
C****      READ TABLE NUMBER AND TITLE
C****
      PRINT 60
60 FORMAT(* ENTER 4 CHARACTERS FOR EACH LABEL FOR Z,Y,X,FXYZ*/
1 * (SEPARATED BY COMMAS)*)
C****
C****      READ TITLES FOR EACH VARIABLE 4 CHARACTERS LONG
C****
      READ 70,LZ,LX,LX,LF
70 FORMAT(4(A4,1X))
C****
C****      GET NUMBER OF 2 VARIABLES AND VALUES
C****
      PRINT 80,LZ
80 FORMAT(* ENTER NUMBER OF *,A4,* VARIABLES--- FREE FORM*)
      CALL GETIN(1,2)
      NZ=Z(1)
      PRINT 90,LZ
90 FORMAT(* ENTER *,A4,* VALUES ,ASCENDING ORDER---FREE FORM*)
      CALL GETIN(NZ,2)
C****
C****      WRITE TO TAPE7 TITLE CARD AND TABLE NUMBER.
C****
100 WRITE(7,10) LNC,LT
      IOPTD=0
      IF(IDFLT.EQ.1)GO TC 99
      PRINT 101
101 FORMAT(* WANT TO SPECIFY DECIMAL PLACES ON TAPE7?*)
      READ 30,ICPTD
89 DO 102 ISET=1,4
102 XV(ISET)=0.
      IF(ICPTD.NE.31B)GO TC 115
      PRINT 110,LZ,LX,LX,LF
110 FORMAT(* ENTER NUMBER OF DECIMAL PLACES FOR *,4(A4,1X)
1 *,* FREE FORM*)
C****
C****      GET NUMBER OF DECIMAL PLACES FOR EACH VARIABLE
C****
      CALL GETIN(4,XV)
115 LZDP=XV(1)
      LYDP=XV(2)
      LXDP=XV(3)
      LFDP=XV(4)
C****
C****      WRITE TO TAPE7 THE 2 VALUES ETC...
C****
      CALL TFORM(1,LZ,NZ,0,0,Z,LZDP,7)
C****
C****      INITIALIZE TEK SCFT:ARE
C****
      CALL INITT(1BR)
      CALL TERM(3,4096)
      CALL CHRSTZ(4)

```

TIGS0038  
 TIGS0039  
 TIGS0040  
 TIGS0041  
 TIGS0042  
 TIGS0043  
 TIGS0044  
 TIGS0045  
 TIGS0046  
 TIGS0047  
 TIGS0048  
 TIGS0049  
 TIGS0050  
 TIGS0051  
 TIGS0052  
 TIGS0053  
 TIGS0054  
 TIGS0055  
 TIGS0056  
 TIGS0057  
 TIGS0058  
 TIGS0059  
 TIGS0060  
 TIGS0061  
 TIGS0062  
 TIGS0063  
 TIGS0064  
 TIGS0065  
 TIGS0066  
 TIGS0067

TIGS0068  
 TIGS0069  
 TIGS0070  
 TIGS0071  
 TIGS0072  
 TIGS0073  
 TIGS0074  
 TIGS0075  
 TIGS0076  
 TIGS0077  
 TIGS0078  
 TIGS0079  
 TIGS0080  
 TIGS0081

TIGS0083  
 TIGS0084  
 TIGS0085  
 TIGS0086  
 TIGS0087  
 TIGS0088

```

      NTITE=0
      DO 250 IZ=1,NZ
      IWIN=0
C****
C****      IF CREATION MCDE THEN SET DEFAULTS TO 0
C****
      IF(IC.NE.1) GO TO 120
      NPTS(1)=0
      X(1)=0
      Y(1)=0
      GO TO 210
C****
C****      NON CREATION MCDE
C****
      120 CONTINUE
      K=1
C****
C****      TRANSFER SECOND INDEPENDENT VARIABLE TO XV ARRAY
C****
      DO 129 J=1,50
      129 XV(J)=0
      DO 130 J=1,NY
      130 XV(J)=A(J)
      LNX=N
      LNY=N
      NPTS(1)=N
      NPTS(2)=0
      IF(IZ.EC.1) GO TO 150
      LNX=0
      LNY=0
      K=0
C****
C****      READ NEXT SET
C****
      150 READ (1,188) LABI,ITI,JTI,KTI,(A(I),I=1,7)
      188 FORMAT(A4,3I1,3X,7F10.4)
      IF(LABI.EQ.4H....)GO TO 189
      IF(LABI.NE.4H...C) GO TO 187
      READ (1,*)IXDEN,IYDEN,ITMX,ITMY,IGRIDY,IGRIDX,AXMIN,AYMIN,
      1AXMAX,AYMAX
      IWIN=1
      GO TO 150
      187 BACKSPACE 01
      GO TO 149
      189 NTITE=NTITE+1
      IF(NTITE.LE.50)GO TO 500
      CALL ANMCDE
      PRINT *,"WARNING 50 LABEL MAX HAS BEEN EXCEEDED UTIGS ABORTED"
      STOP
      500 READ (1,191) (ITC(I,NTITE),I=1,8)
      191 FORMAT(8A10)
C****
C**** ITT= FLAG =1 TITLE NOT CENTERED =0 TITLE IS CENTERED
C**** JTI= UNITS FLAG =1 USE REAL UNITS =0 USE SCREEN UNITS
C**** KTI= FLAG KTI=1 TITLE IS AN ARROW
C**** THE NEXT 6 INPUTS ARE 3 PSIRS OF COORDINATES
C**** THE FIRST = X,Y CENTERED PAIR
C**** THE SECOND IS A VECTOR USED FOR THE SLOPE
C**** THE THIRD IS ONLY USED FOR THE ARROW
C**** IF ARROW THE SECOND IS USED AS THE SECOND (KNEE) POINT
C**** OF THE ARROW.
      ITT(NTITE)=ITI

```

TIGS0089

TIGS0090

TIGS0091

TIGS0092

TIGS0093

TIGS0094

TIGS0095

TIGS0096

TIGS0097

TIGS0098

TIGS0099

TIGS0100

TIGS0101

TIGS0102

TIGS0103

TIGS0104

TIGS0105

TIGS0106

TIGS0107

TIGS0111

TIGS0112

TIGS0113

TIGS0114

TIGS0115

TIGS0116

TIGS0117

```

      JTT(NTITE)=JTI
      KTT(NTITE)=KTI
      STT(NTITE)=A(1)
      XTT(NTITE)=A(2)
      YTT(NTITE)=A(3)
      XU(NTITE)=A(4)
      YUU(NTITE)=A(5)
      XVV(NTITE)=A(6)
      YVV(NTITE)=A(7)
      GC TO 150
149 CALL TABR(LK,L,MT,K,A,1)
C****
C****      CHECK FOR NEXT 2 GROUP
C****
      IF(LW.NE.LY) GC TO 151
      NY=N
      GO TO 210
C****
C****      CHECK FOR END OF TABLE
C****
151 IF(LW.EQ.4HECT ) GC TO 210
C****
C****      CHECK FOR NEXT X DATA
C****
      IF(LW.NE.LX) GO TO 170
C****
C****      DATA IS X DATA STORE IT
C****
      JXS=LNK
      DO 160 J=1,K
      LNX=LNK+1
160 X(LNX)=A(J)
      GC TO 150
C****
C****      DATA HAD BETTER BE LY
C****
170 IF(LW.NE.LF) STOP
C****
C****      IF DATA HAS NOT BE INPUT FOR X DATA USE LAST VALUES
C****
      IF(LNX.GT.LNY) GO TO 190
      LJS=JXS
      DO 180 J=1,K
      LNX=LNX+1
      LJS=LJS+1
180 X(LNX)=X(LJS)
C****
C****      UPDATE COUNTERS
C****
190 K=K+1
      NPTS(K)=N
      NPTS(K+1)=0
C****
C****      LOAD Y DATA
C****
C**** CHECK FOR 50 LINE LIMIT
C****
      IF(K.NE.49) GC TO 202
      CALL ANMODE
      PRINT *,"NUMBER OF LINES EXCEED 49 UTIGS ABORTED"
      STOP

```

TIGS0119  
TIGS0120  
TIGS0121  
TIGS0122

TIGS0123  
TIGS0124  
TIGS0125  
TIGS0126  
TIGS0127  
TIGS0128  
TIGS0129  
TIGS0130  
TIGS0131  
TIGS0132  
TIGS0133

TIGS0135  
TIGS0136  
TIGS0137  
TIGS0138  
TIGS0139  
TIGS0140  
TIGS0141  
TIGS0142  
TIGS0143  
TIGS0144  
TIGS0145  
TIGS0146

TIGS0148  
TIGS0149

TIGS0151  
TIGS0152  
TIGS0153  
TIGS0154  
TIGS0155  
TIGS0156  
TIGS0157  
TIGS0158  
TIGS0159  
TIGS0160

```

202 IF(LNY+N.LE.1000) GO TO 199
    CALL ANKMODE
    PRINT *, "NUMBER OF POINTS EXCEED 1000 LIMIT UTIGS ABCRTEED"
    STOP
199 DO 200 J=1,N
    LNY=LNY+1
200 Y(LNY)=A(J)
    MTT(K)=MT
    LTT(K)=L
C****
C****      GC BACK TO GET NEXT GROUP
C****
    GC TO 150
C****
C****      PLOT DATA
C****
210 IGRID=L
    CALL TIGPPR(NPLOT,LF,1,LX,1,LT,7,X,Y,NPTS,LY,1,XV,LYDP,ITIP,
1 IGRID,LZ,Z(IZ))
    CALL ANKMODE
C****
C****      COUNT NUMBER OF Y VALUES
C****
    JY=0
    DO 220 I=1,50
    IF(NPTS(I).EQ.0) GO TO 230
    JY=JY+1
220 CONTINUE
    GO TO 250
C****
C****      WRITE TO TAPE7 Y DATA ETC...
C****
230 CALL TFORM(1,LY,JY,0,0,XV,LYDP,7)
    LCC=1
    J=0
240 J=J+1
    NP=NPTS(J)
    IF(NP.EQ.0) GO TO 250
C****
C****      WRITE TO TAPE7 X DATA ETC...
C****
    CALL TFCRM(LCC,LX,NP,0,0,X,LXDP,7)
C****
C****      WRITE TO TAPE7 Y DATA ETC...
C****
    CALL TFCRM(LCC,LF,NP,LTT(J),MTT(J),Y,LYDP,7)
    LOC=LCC+NP
    GO TO 240
250 CONTINUE
    WRITE(7,188)4H...C
    WRITE(7,*)IXDEN,IYDEN,ITMX,ITMY,IGRIDY,IGRIDX,AXMIN,AYMIN,
1AXMAX,AYMAX
C*** CHECK FOR TITLES IN FILE
    IF(NTITE.LT.1) GO TO 310
    LAEI=4H....
    DO 300 I=1,NTITE
    WRITE(7,188)LABI,ITT(I),JTT(I),KTT(I),SIT(I),XTT(I),YTT(I),XUU(I),
1 YUU(I),XVV(I),YVV(I)
300 WRITE(7,191) (ITC(J,I),J=1,8)
310 WRITE (7,10) IEND
C****
C**** IF NON CREATION MODE THEN GO BACK TO READ NEXT TITLE

```

TIGS0161  
TIGS0162  
TIGS0163

TIGS0164  
TIGS0165  
TIGS0166  
TIGS0167  
TIGS0168  
TIGS0169  
TIGS0170

TIGS0171  
TIGS0172  
TIGS0173  
TIGS0174  
TIGS0175  
TIGS0176  
TIGS0177  
TIGS0178  
TIGS0179  
TIGS0180  
TIGS0181  
TIGS0182  
TIGS0183  
TIGS0184  
TIGS0185

TIGS0187  
TIGS0188  
TIGS0189  
TIGS0190  
TIGS0191  
TIGS0192  
TIGS0193  
TIGS0194

TIGS0196  
TIGS0197  
TIGS0198

TIGS0200  
TIGS0201  
TIGS0202

TIGS0203

C\*\*\*\*

```

      IF(IC.NE.1) GO TO 31
251 WRITE (7,10)
      REWIND 7
      END

```

CTFORM

TIGS0204  
TIGS0205  
TIGS0206

```

      SUBROUTINE TFCRM(LCC,LAP,N,L,LM,X,IPI,K)
      DIMENSION X(1),IFCRM(4)

```

TFCR0001

```

C****      FORMATTING SUBROUTINE FOR TPLLOT FORMAT
C****      LCC IS THE LCCAL ARRAY POSITION TO PRINT FROM
C****      LAB IS THE 4 CHARACTER LABEL
C****      N IS THE NUMBER TO PRINT
C****      X IS THE ARRAY CONTAINING THE VALUES
C****      IP IS THE NUMBER OF DECIMAL PLACES TO USE IN FORMAT
C****      IF IP=0 THEN MAXIMUM NUMBER OF DECIMAL PLACES WILL RESULT

```

TFCR0003  
TFCR0004  
TFCR0005  
TFCR0006  
TFCR0007  
TFCR0008  
TFCR0009

```

      IP=IPI
      IF(IP.NE.0)GO TO 20
      XMAX=0.
      IF(N.LE.1)GO TO 20
      DO 10 I=1,N
      IF(X(I).EQ.0.)GO TO 10
      XC=ALOG10(ABS(X(I)))
      IF(XC.GT.XMAX) XMAX=XC
10 CONTINUE
      IP=7-IFIX(XMAX)
20 IF(IP.LT.0) IP=0
      IF(IP.GT.7) IP=7
      JO=LCC-1
      IFCRM(1)=10H(A4,I3.
      IFORM(3)=24343456420634335733E+IP
      IFCRM(4)=10H))
      IF((L+LM).EQ.0)GO TO 30
      IFORM(2)=10HI1,I2,(
      WRITE(K,IFORM) LAP,N,L,LM,(X(I+JO),I=1,N)
      RETURN
30 IFCRM(2)=10H 3X,(
      WRITE(K,IFORM) LAP,N,(X(I+JO),I=1,N)
      RETURN
      END

```

TFCR0010  
TFCR0011  
TFCR0012

TFCR0016

TFCR0021  
TFCR0022

CTAER

TAER0001

```

      SUBROUTINE TABR(LAP,L,MT,N,A,K)
      DIMENSION A(1)
      READ(K,10) LAB,N,L,MT,(A(I),I=1,7)
10 FORMAT(A4,I3,I1,I2,7F10.0)
      IF(N.GT.7) READ(K,20) (A(I),I=8,N)
20 FORMAT(10X,7F10.0)
      RETURN
      END

```

CTIGP

TIGP0001

C  
C  
C

TEK INTERACTIVE GPPR M CADDY FEE 78

TIGP0002  
TIGP0003  
TIGP0004  
TIGP0005  
TIGP0006

```

      SUBROUTINE TIGPPR(NPLOT,LAEY,N1,LAEY,N2,LAETL,NT,X,Y,
1 NPTA,LAEVAL,NCC,VLAEL,NDECVIN,ITIP,IGRID,LZ,ZVAL)
      COMMON/ITITLE/ITT(50),JTT(50),KTT(50),LTT(50),MTT(50),STT(50),
1 XTT(50),YTT(50),XUU(50),YUU(50),XVV(50),YVV(50),ITC(8,50),NTITE
2 ,AXMIN,AYMIN,AXMAX,AYMAX,IXDEN,IYDEN,IGRIDX,IGRIDY,ITMX,ITMY,
3 ICHAR,IWIN
      COMMON/TKTRNX/ITEKC(60)

```

TIGP0009

```

DIMENSION X(200),Y(200),LABTL(7),NPTA(50),VLABEL(50),
1 LABX(5),LABY(5),VTEM(8),LABVAL(8),IQUICK(30),ISUB(1000),MSG7(29)
DIMENSION MSG1(20),MSG2(22),MSG4(10),MSG5(10),MSG6(15),IALTM(6,2)
DIMENSION IPPX(4,50),IPPY(4,50),XDUM(2)
EQUIVALENCE (BEG(1),XBEG),(BEG(2),YBEG)
EQUIVALENCE (DEL(1),DELX),(DEL(2),DELY),(ITAB,LTV(2))
EQUIVALENCE (EN(1),XEND),(EN(2),YEND)
EQUIVALENCE (IOFF,ITEKC(30)),(TXMIN,ITEKC(1))
C      SET LINE SPACING
COMMON/TEKGPPR/LDEL,LCNT,MAXSR,LTV(17),EN(2),DEL(2),BEG(2),RDX2,
1RDY2,NLINE,NDRAW(50)
DATA MSG1/46,46,80,111,105,110,116,101,114,32,
1      80,111,115,105,116,105,111,110,101,100/
DATA MSG2/73,110,112,117,116,32,76,105,110,101,
1      32,86,97,108,117,101,44,76,84,44,77,84/
C**** ILLEGAL MESSAGE
DATA MSG4/42,73,106,108,101,103,97,108,42,32/
DATA MSG5/73,110,112,117,116,32,88,44,89,32/
DATA MSG6/65,32,111,114,32,66,32,109,111,100,
1      101,63,32,32,32/
DATA MSG7/69,78,84,65,82,32,87,73,78,68,79,87,13,10,
1      68,73,65,71,79,78,65,76,32,80,79,73,78,84,83/
DATA ((IALTM(I,J),I=1,6),J=1,2)/65,102,116,101,114,32,
1      66,101,102,111,114,101/
DATA IQUICK/0,0,1,1,2,3,10,4,0,0,
1      11,12,0,5,0,6,0,7,8,0,
2      0,0,9,0,0,0,0,0,0,0/
IXTOS(X)=((X-BEG(1))/DELX+600)
IYTOS(Y)=((Y-BEG(2))/DELY+300)
LDEL=50
ICHAH=0
IF(IWIN.EQ.1)GO TO 5
IGRIDX=IGRID
IGRIDY=IGRID
C**** DEFAULT AXIS
IXDEN=6
IYDEN=8
ITMX=0
ITMY=0
5 IF(NPLOT.GT.0) GO TO 20
DO 10 I=1,8
10 LTV(I)=0
C**20 IWIN=0
20 NPLOT=NPLOT+1
KLAST=0
NTL=NT
NLINE=0
30 NSTOR=0
C      SET STORAGE POINTER TO INITIAL SEQUENCE
DO 40 I=1,999
40 ISUB(I)=I+1
C
C      MERGE HERE TO REPLOT
C
50 CALL SWCHAR(1)
LCNT=3120
C      SUM UP NUMBER OF POINTS
NL=0
NPTOT=0
DO 60 I=1,50
N=NPTA(I)
IF(N.EQ.0) GO TO 70

```

```

TIGP0010
TIGP0011
TIGP0012
TIGP001
TIGP0014
TIGP0015
TIGP0016
TIGP0017
TIGP0018
TIGP0019
TIGP0020
TIGP0021
TIGP0022
TIGP0023
TIGP0024
TIGP0025
TIGP0026
TIGP0027
TIGP0028
TIGP0029
TIGP0030
TIGP0031
TIGP0032
TIGP0033
TIGP0034
TIGP0035
TIGP0036
TIGP0037
TIGP0038
TIGP0039
TIGP0040
TIGP0041
TIGP0042
TIGP0101
TIGP0102
TIGP0043
TIGP0044
TIGP0045
TIGP0046
TIGP0047
TIGP0048
TIGP0049
TIGP0050
TIGP0051
TIGP0052
TIGP0053
TIGP0054
TIGP0055
TIGP0056
TIGP0057
TIGP0059
TIGP0060
TIGP0061
TIGP0062
TIGP0063
TIGP0064
TIGP0065

```

```

      NL=NL+1
      NPTOT=NPTOT+N
C      SET STORAGE LIMIT TO NPTOT FIRST PASS
      70 IF(NSTOR.EQ.0) NSTOR=NPTOT
      IF(NPTOT.GT.0)GO TO 90
      NSTOR=0
      CALL MOVABS(0,LCNT)
      CALL ANMODE
      PRINT 80
      80 FCRMAT(* NO DATA FOUND TO PLOT ..ENTER COMMAND*)
      LCNT=LCNT-LDEL
      IPLOT=0
      GO TO 280
      90 CONTINUE
      IDEN=0
C      PREPARE TEKTRONIX AGII COMMON
C
      IPLOT=IPLOT+1
      CALL CHRSTZ(2)
      IF(ICHR.NE.0) GO TO 120
      CALL ANMODE
      PRINT 110,LABTL
      110 FORMAT(* NEXT PLOT IS *,/,9A10)
      IF(IWIN.NE.0)PRINT *,"GRID/WINDOW IS SET "
      1,IXDEN,IYDEN,ITMX,ITMY,IGRIDY,IGRIDX,AXMIN,AYMIN,
      2 AXMAX,AYMAX
      LCNT=LCNT-LDEL*3
C****
C
C      IF FIRST TIME AROUND.. GET COMMAND FIRST TO PLOT
C
C      SET SCREEN WINDOW SIZE
      120 CALL SWINDO(600,3360,300,2400)
      IF(IWIN.EQ.0) GO TO 125
C**** THIS SETS KLAST PARAMETER IF NOT SET
      IF(KLAST.NE.0)GO TO 140
      K=1
      DO 121 I=1,NPTOT
      KLAST=K
      121 K=ISUE(K)
      GO TO 140
      125 AXMAX=-1.E99
      AYMAX=-1.E99
      AXMIN=+1.E99
      AYMIN=+1.E99
C      SET MIN AND MAX DATA VALUES
      K=1
      DO 130 I=1,NPTOT
      AXMIN=AMIN1(AXMIN,X(K))
      AYMIN=AMIN1(AYMIN,Y(K))
      AXMAX=AMAX1(AXMAX,X(K))
      AYMAX=AMAX1(AYMAX,Y(K))
      KLAST=K
C      SET KLAST TO END STORAGE VALUE
      130 K=ISUE(K)
      IWIN=1
      140 IF(AXMIN.NE.AXMAX) GO TO 150
      AXMIN=AXMIN-.5
      AXMAX=AXMAX+.5
      150 IF(AYMIN.NE.AYMAX) GO TO 160
      AYMIN=AYMIN-.5
      AYMAX=AYMAX+.5

```

TIGP0066  
 TIGP0067  
 TIGP0068  
 TIGP0069  
 TIGP0070  
 TIGP0071  
 TIGP0072  
 TIGP0073  
 TIGP0074  
 TIGP0075  
 TIGP0076  
 TIGP0077  
 TIGP0078  
 TIGP0083  
 TIGP0085  
 TIGP0086  
 TIGP0087  
 TIGP0089  
 TIGP0093  
 TIGP0094  
 TIGP0095  
 TIGP0096  
 TIGP0097

TIGP0098  
 TIGP0099  
 TIGP0103  
 TIGP0104  
 TIGP0105  
 TIGP0106  
 TIGP0110

TIGP0111  
 TIGP0112  
 TIGP0113  
 TIGP0114  
 TIGP0115  
 TIGP0116  
 TIGP0117  
 TIGP0118  
 TIGP0119  
 TIGP0120  
 TIGP0121  
 TIGP0122  
 TIGP0123  
 TIGP0124  
 TIGP0125  
 TIGP0126  
 TIGP0127  
 TIGP0128  
 TIGP0129  
 TIGP0130  
 TIGP0131



```

160 CONTINUE
   XDUM(1)=AXMIN
   XDUM(2)=AXMAX
   AXL=IXDEN
   CALL AXSCALE(XDUM,AXL,2,XBEG,DELXX,0)
   XEND=XBEG+AXL*DELXX
   XDUM(1)=AYMIN
   XDUM(2)=AYMAX
   AXL=IYDEN
   CALL AXSCALE(XDUM,AXL,2,YBEG,DELYY,0)
   YEND=YBEG+AXL*DELYY

C
C   FIND VIRTUAL SPACE TO SCREEN SPACE SCALING PARAMETERS
C
   DELX=(XEND-XBEG)/3360.
   DELY=(YEND-YBEG)/2400.
   RDX2=1./(DELX*DELX)
   RDY2=1./(DELY*DELY)

C   SET VIRTUAL WINDOW
   CALL DWINDO(XBEG,XEND,YBEG,YEND)
   IF(ICHAR.EQ.0)GO TO 280
   CALL DRAWIT(NL,NPTA,X,Y,ISUE,MTT,LTT)
   CALL CHRSTZ(4)
   IF(IXDEN.LE.0)GO TO 172
   CALL AXIS
   1(600,300,3360,2400,IXDEN,0,XBEG,DELXX,EX,1,1,ITMX,0,.15)
172 IF(IYDEN.LE.0)GO TO 174
   CALL AXIS
   1(600,300,3360,2400,IYDEN,1,YBEG,DELYY,EX,1,1,ITMY,0,.15)
174 CALL SWCHAR(1)
   ICH=KIN(.15*.873)
   ICV=1.6*ICH
   CALL PLCHAR(ICH,ICV)
   IF(IXDEN*IYDEN.EQ.0)GO TO 190
   CALL PTITE(2280,3000,NT,LABTL,0,50,0.,IPPX,IPPY)
   IF(NTITE.GT.1) GO TO 180
   CALL PTITE(300,1500,N1,LAEX,0,50,90.,IPPX,IPPY)
180 IF(NTITE.GE.1) GO TO 190
   CALL PTITE(2280,100,N2,LAEX,0,20,0.,IPPX,IPPY)
190 IF(NTITE.EQ.0)GO TO 250
   DO 240 II=1,NTITE
   IF(KTT(II).EQ.0)GO TO 210
C**** ARROW CODE
C**
   IA1=IXTOS(XUU(II))
   IA2=IYTOS(YUU(II))
   IB1=IXTOS(XVV(II))
   IB2=IYTOS(YVV(II))
   CALL MOVEA(XTT(II),YTT(II))
   CALL DARRCW(IA1,IA2,IB1,IB2)
   DO 200 JJ=1,4
   IPPX(JJ,II)=1000000000
200 IPPY(JJ,II)=1000000000
   GO TO 240
210 DANGX=(XVV(II)-XUU(II))*SQRT(RDX2)
   DANGY=(YVV(II)-YUU(II))*SQRT(RDY2)
   ANG=0.
   IF(DANGY.EQ.0..AND.DANGX.EQ.0.) GO TO 211
   ANG=ATAN2(DANGY,DANGX)*57.2957795
C**** DEFAULT SIZE TO BE .15 IF NEGATIVE OR ZERO
211 IF(STT(II).LE.0.)STT(II)=.15
   ICH=KIN(STT(II)*.873)
   ICV=ICH*1.6

```

TIGP0158  
TIGP0159  
TIGP0160  
TIGP0156  
TIGP0157  
TIGP0161  
TIGP0162  
TIGP0132

TIGP0164  
TIGP0169  
TIGP0168

TIGP0170

TIGP0172  
TIGP0173

TIGP0174  
TIGP0175  
TIGP0176  
TIGP0177  
TIGP0178  
TIGP0179  
TIGP0180

TIGP0181  
TIGP0182  
TIGP0183  
TIGP0184  
TIGP0185  
TIGP0186  
TIGP0187  
TIGP0188  
TIGP0189  
TIGP0190  
TIGP0191  
TIGP0192  
TIGP0193

TIGP0195  
TIGP0196

TIGP0198

IF(ITT(II).EQ.1)GO TO 220	TIGP0199
IX=IXTOS(XTT(II))	TIGP0200
IY=IYTOS(YTT(II))	TIGP0201
GO TO 230	TIGP0202
220 IX=XTT(II)	TIGP0203
IY=YTT(II)	TIGP0204
230 CALL PLCHAR(ICH,ICV)	TIGP0205
IC=JTT(II)	TIGP0206
IA=KTT(II)	TIGP0207
CALL PTITE(IX,IY,2,ITC(1,II),IC,40,ANG,IPPX(1,II),IPPY(1,II))	TIGP0208
240 CONTINUE	TIGP0209
CALL LINRGT(0.)	TIGP0210
250 NYM=IYDEN	
NXM=IXDEN	
CALL SHADEIT(IPPX,IPPY,NTITE)	
IF(IGRIDX.EQ.0)GO TO 260	TIGP0211
CALL PGRIDX(IPPX,IPPY,NTITE,NXM,NYM,1000,IGRIDX)	
260 IF(IGRIDY.EQ.0)GO TO 270	
CALL PGRIDY(IPPX,IPPY,NTITE,NXM,NYM,1000,IGRIDY)	
C	TIGP0217
C	TIGP0218
C	TIGP0219
270 CONTINUE	TIGP0220
280 CALL SWCHAR(0)	
CALL CHRSLZ(4)	TIGP0221
IKILL=0	TIGP0222
IF(LCNT.LT.220) GO TO 710	TIGP0223
IF(NPTOT.EQ.1) GO TO 320	TIGP0224
CALL GETVAL(ICHAR,X0,Y0)	TIGP0225
290 IF(ICHAR.LE.64.OR.ICHAR.GE.95)GO TO 300	TIGP0226
ICHAR=ICHAR-64	TIGP0227
ICHECK=IQUICK(ICHAR)	TIGP0228
IF(ICHECK.EQ.0) GO TO 300	TIGP0229
GO TO (390,500,540,280,590,640,720,740,770,570,800,790),ICHECK	TIGP0230
300 LCNT=LCNT-LDEL	TIGP0231
CALL NOTATE(0,LCNT,10,MSG4)	TIGP0232
GO TO 280	TIGP0233
C	TIGP0234
C	TIGP0235
C	TIGP0236
C	TIGP0237
C	TIGP0238
310 IF(NPTOT.EQ.0)GO TO 590	TIGP0239
320 LCNT=LCNT-LDEL	TIGP0240
CALL NOTATE(0,LCNT,20,MSG1)	TIGP0241
330 CALL GETVAL(ICHAR,X0,Y0)	TIGP0242
C	TIGP0243
C	TIGP0244
C	TIGP0245
340 IF(ICHAR.EQ.65) GO TO 350	TIGP0246
C	TIGP0247
C	TIGP0248
C	TIGP0249
IF(ICHAR.EQ.66) GO TO 290	TIGP0250
350 CALL POINTA(X0,Y0)	TIGP0251
CALL MOVEA(X0,Y0)	TIGP0252
IF(IOFF.EQ.0)CALL ANCHO(IS)	TIGP0253
IF(ICHAR.EQ.77) GO TO 370	TIGP0254
C*** CHECK FOR STORAGE EXCEEDING 280 WARNING LIMIT	TIGP0255
C	TIGP0256
C	TIGP0257
C	TIGP0258

C	NSTOR=NSTOR+1	TIGP0259
C***	NEW LIST LINKED DATA STORAGE	
C***	NSTOR = LAST USED CELL	
C***	NPTOT = NUMBER OF ACTUAL POINTS	
	ISTOR=ISUB(KLAST)	
	ISUB(KLAST)=ISUB(ISTOR)	
	IF(JSAVE.EQ.KLAST)KLAST=ISTOR	
	NSTOR=NSTOR+1	
	NPTOT=NPTOT+1	
	NPTA(ISAVE)=NPTA(ISAVE)+1	TIGP0260
C	MOVE POINTER OF CLOSEST POINT TO END	TIGP0261
	ISUB(ISTOR)=ISUB(JSAVE)	TIGP0262
C	CHANGE CLOSEST POINTER TO ACCESS LAST POINT	TIGP0263
	ISUB(JSAVE)=ISTOR	TIGP0264
	IF(ICHAR.EQ.65) GO TO 360	TIGP0265
C	MOVE OLD POINT TO LAST POINT ( INSERT BEFORE)	TIGP0266
	X(ISTOR)=X(JSAVE)	TIGP0267
	Y(ISTOR)=Y(JSAVE)	TIGP0268
	GO TO 370	TIGP0269
C	NEW POINT ADD AFTER	TIGP0270
	360 JSAVE=ISTOR	TIGP0272
	370 X(JSAVE)=X0	TIGP0273
	Y(JSAVE)=Y0	TIGP0274
	IF(NSTOR.LT.995) GO TO 330	TIGP0275
	CALL NEWPAG	TIGP0276
	CALL CHRSLZ(1)	TIGP0277
	CALL ANMODE	TIGP0278
	PRINT 380, NSTOR-NPTOT+5	TIGP0279
	380 FORMAT(* WARNING DATA STORAGE IS NEARING 1000 MAX LIMIT*/	TIGP0280
	1 * A REPACK PROCEDURE HAS BEEN INVOKED TO GIVE YOU*,13,/*	TIGP0281
	2 * MORE STORAGE LOCATIONS....PRESS RETURN TO CONTINUE*	TIGP0282
	3 ,/,* AND DELETE POINTS IF YOU CAN*)	TIGP0283
	CALL TINPUT(I)	TIGP0284
	CALL CHRSLZ(4)	TIGP0285
	GO TO 500	TIGP0286
C		TIGP0287
C	DELETE POINT (D)	TIGP0288
C		TIGP0289
	390 DSAVE=1.E40	TIGP0290
	IF(NPTOT.EQ.0) GO TO 280	TIGP0291
	IS=64	TIGP0292
	NSUM=1	TIGP0293
	K=1	TIGP0294
	DO 430 I=1,NL	TIGP0295
	NEND=NSUM+NPTA(I)-1	TIGP0296
C***		TIGP0297
C***	SAVE THE POINT BEGINNING EACH LINE	TIGP0298
	KBEGN=K	TIGP0299
	DO 420 J=NSUM,NEND	TIGP0300
	IF(IKILL.LE.0)GO TO 395	
	IF(I.NE.IKILL)GO TO 410	
395	IF(NLINE.EQ.0) GO TO 400	TIGP0301
	IF(NDRAW(I).EQ.0) GO TO 410	TIGP0302
400	DX=X(K)-X0	TIGP0303
	YDY=Y(K)-Y0	TIGP0304
	DIST=DX*DX*RDY2+YDY*YDY*RDY2	TIGP0305
	IF(DIST.GE.DSAVE)GO TO 410	TIGP0306
	DSAVE=DIST	TIGP0307
	JSAVE=K	TIGP0308
	ISAVE=I	TIGP0309
	KLINE=KBEGN	TIGP0310
410	KBEFL=KLAST	

```

      KLAST=K
420 K=ISUB(K)
430 NSUM=NSUM+1
      IS=ISAVE+64
440 IF(IS.LE.90)GO TO 450
      IS=IS-90
      GO TO 440
450 CALL POINTA(X(JSAVE),Y(JSAVE))
      IF(ICFF.EQ.0)CALL ANCHO(IS)
      IF(ICHAR.EQ.11)GO TO 810
      IF(ICHAR.NE.4) GO TO 310
      K=JSAVE
C**** NEW REVISION SPRING 82
      IF(K.NE.KLAST) GO TO 460
      KLAST=KBEFL
      GO TO 470
460 KFW=ISUB(K)
      IF(KFW.NE.KLAST)GO TO 462
      KLAST=K
      GO TO 465
462 ISUB(K)=ISUB(KFW)
      ISUB(KFW)=ISUB(KLAST)
      ISUB(KLAST)=KFW
465 X(K)=X(KFW)
      Y(K)=Y(KFW)
470 NPTOT=NPTOT-1
      NSTOR=NSTOR-1
      NPTA(ISAVE)=NPTA(ISAVE)-1
      IF(NPTA(ISAVE).GT.0)GO TO 490
      NPTA(ISAVE)=0
      J=0
      DO 480 I=1,NL
      IF(I.EC.ISAVE)GO TO 480
      J=J+1
      NPTA(J)=NPTA(I)
      VLABEL(J)=VLABEL(I)
      MTT(J)=MTT(I)
      LTT(J)=LTT(I)
480 CONTINUE
      NPTA(NL)=0
      NL=NL-1
      IKILL=0
490 IF(IKILL.EQ.0)GO TO 280
      IKILL=ISAVE
      GO TO 810

C
C
C
500 CALL NEW-PAG
      L=1
      DO 530 I=2,NPTOT
      K=ISUB(L)
      IF(I.EC.K) GO TO 530
      J=K
      JLEFT=NPTOT+1-I
      DO 510 KK=1,JLEFT
      IF(J.EC.I) GO TO 520
      JO=J
510 J=ISUB(JO)
520 ISUB(JO)=K
      ISUB(L)=I
      IS=ISUB(I)

```

TIGP0311  
TIGP0312  
TIGP0313  
TIGP0314  
TIGP0315  
TIGP0316  
TIGP0317  
TIGP0318  
TIGP0319  
TIGP0320  
TIGP0321  
TIGP0323

TIGP0338  
TIGP0339  
TIGP0340  
TIGP0341  
TIGP0342  
TIGP0343  
TIGP0344  
TIGP0345  
TIGP0346  
TIGP0347  
TIGP0348  
TIGP0349  
TIGP0350  
TIGP0351  
TIGP0352  
TIGP0353

TIGP0354  
TIGP0355  
TIGP0356  
TIGP0357  
TIGP0358  
TIGP0359  
TIGP0360  
TIGP0361  
TIGP0362  
TIGP0363  
TIGP0364  
TIGP0365  
TIGP0366  
TIGP0367  
TIGP0368  
TIGP0369  
TIGP0370  
TIGP0371

```

      ISUB(I)=ISUB(K)
      ISUB(K)=IS
      XS=X(I)
      X(I)=X(K)
      X(K)=XS
      XS=Y(I)
      Y(I)=Y(K)
      Y(K)=XS
530 L=I
      IF(ICHECK.EQ.2) RETURN
      GO TO 30
C
C      FORMAT (F)
C
540 LCNT=LCNT-LDEL
      CALL MOVABS(0,LCNT)
      CALL ANMCDE
      IY=(Y0-YEEG)/DELY+300
      II=(3045-IY)/50+1
      IF(II.LT.1) II=1
      IF(II.GT.NL) II=NL
      PRINT 550,LTT(II),MTT(II)
550 FORMAT(* LT= *,I2,* MT= *,I2)
      CALL GETIN(2,VTEM)
      LTT(II)=VTEM(1)
      MTT(II)=VTEM(2)
      LCNT=LCNT-LDEL
C
C      IF F OUTSIDE OF AXIS THE SET ALL CURVE NCDES
C
      IF(X0.LE.TMX.IN) GO TO 280
      DO 560 I=1,50
      LTT(I)=VTEM(1)
560 MTT(I)=VTEM(2)
      GO TO 280
C
C      CHANGE GRID OPTION
C
570 LCNT=LCNT-LDEL
      CALL MOVABS(0,LCNT)
      CALL ANMCDE
      PRINT *,
1"GRID OPTIONS,(IN X,Y PAIRS)MAJOR,MINOR TICK MARKS , GRID SWTCHS"
      CALL GETIN(6,VTEM)
      LCNT=LCNT-LDEL
      IXDEN=VTEM(1)
      IYDEN=VTEM(2)
      ITMX=VTEM(3)
      ITMY=VTEM(4)
      IGRIDY=VTEM(5)
      IGRIDX=VTEM(6)
      GO TO 280
C
C      NEW LINE (N)
C
590 NPTOT=NPTOT+1
      NSTOR=NSTOR+1
      IF(NPTOT.NE.1) GO TO 591
      Istor=1
      GO TO 592
591 Istor=ISUB(KLAST)
592 KLAST=ISTOR

```

TIGP0372  
 TIGP0373  
 TIGP0374  
 TIGP0375  
 TIGP0376  
 TIGP0377  
 TIGP0378  
 TIGP0379  
 TIGP0380  
 TIGP0381  
 TIGP0382  
 TIGP0383  
 TIGP0384  
 TIGP0385  
 TIGP0386  
 TIGP0387  
 TIGP0388  
 TIGP0389  
 TIGP0390  
 TIGP0391  
 TIGP0392  
 TIGP0393  
 TIGP0394  
 TIGP0395  
  
 TIGP0396  
 TIGP0397  
 TIGP0398  
 TIGP0399  
 TIGP0400  
 TIGP0401  
 TIGP0402  
  
 TIGP0403  
 TIGP0404  
 TIGP0405  
 TIGP0406  
 TIGP0407  
 TIGP0408  
 TIGP0409  
 TIGP0410  
  
 TIGP0413  
  
 TIGP0414  
 TIGP0415  
  
 TIGP0416  
 TIGP0417  
 TIGP0420  
 TIGP0421  
 TIGP0422  
 TIGP0423  
 TIGP0424  
 TIGP0425  
 TIGP0426

```

      IF(IPL0T+ITAB.EQ.0)GO TC 760
600 X(ISTOR)=X0
      Y(ISTOR)=Y0
      NL=NL+1
      IS=NL+64
      ISAVE=NL
      JSAVE=ISTOR

C
C      NO DATA THEN DO(T SYMBOL IT
C
      IF(IPL0T.EQ.0) GO TC 630
610 IF(IS.LE.90)GO TO 620
      IS=IS-90
      GO TC 610
620 CALL POINTA(X0,Y0)
      IF(IGFF.EQ.0)CALL ANCHO(IS)
630 NPTA(NL)=1
      NL1=NL+1
      NPTA(NL1)=0
      LCNT=LCNT-LDEL
      CALL NCTATE(0,LCNT,22,MSG2)
      LCNT=LCNT-LDEL
      CALL MCVABS(0,LCNT)
      CALL ANMODE
      CALL GETIN(3,VTEM)
      VLABL(NL)=VTEM(1)
      LTT(NL)=VTEM(2)
      MTT(NL)=VTEM(3)
C*****ISAVE=NL
C*****JSAVE=NSTOR
      IF(IPL0T.EQ.0) GO TO 710
      GO TO 320

C
C      PLOT (P)
C
C
C
C      CHECK FOR TABLET MODE, SKIP SPECIAL P SECTION IF TABLET
C
640 IF(ITAB.EQ.1) GO TO 710
      IF(X0.GT.TXMIN) GO TO 710
      IF(IDEN.GT.0)GO TO 680
      LCNT=3120
      CALL MCVABS(0,LCNT)
      CALL ANMODE
      PRINT 650,(LAEVAL(J1),J1=1,NCC)
      CALL MCVABS(2800,2800)
650 FORMAT(8A10)
      LCNT=LCNT-LDEL
      KL=0
      DO 670 J1=1,NL
      LCNT=LCNT-LDEL
      CALL MCVABS(0,LCNT)
      IF(KL.EQ.27)KL=0
      KL=KL+1
      CALL ANMODE
      PRINT 660,KL,VLABL(J1)
660 FORMAT(1X, R1,G13.5)
670 CONTINUE
      IDEN=1
C**** GET BACK TO PLOTTING LABEL ONLY TURNED ON
      GO TO 280
C****

```

TIGP0428  
 TIGP0429  
 TIGP0430  
 TIGP0431  
 TIGP0432

TIGP0433  
 TIGP0434  
 TIGP0435  
 TIGP0436  
 TIGP0437  
 TIGP0438  
 TIGP0439  
 TIGP0440  
 TIGP0441  
 TIGP0442  
 TIGP0443  
 TIGP0444  
 TIGP0445  
 TIGP0446  
 TIGP0447  
 TIGP0448  
 TIGP0449

TIGP0451  
 TIGP0452  
 TIGP0453  
 TIGP0454  
 TIGP0455  
 TIGP0456  
 TIGP0457  
 TIGP0458  
 TIGP0459  
 TIGP0460  
 TIGP0461  
 TIGP0462  
 TIGP0463

TIGP0465  
 TIGP0466  
 TIGP0467  
 TIGP0468  
 TIGP0469  
 TIGP0470  
 TIGP0471  
 TIGP0472  
 TIGP0473  
 TIGP0474  
 TIGP0475  
 TIGP0476  
 TIGP0477  
 TIGP0478  
 TIGP0479  
 TIGP0480

```

680 IF(NLINE.GT.0) GO TO 700
    DO 690 I=1,NL
690 NDRAW(I)=0
700 IY=(Y0-YBEG)/DELY+300
    II=(3045-IY)/50+1
    IF(II.LT.1) II=1
    IF(II.GT.NL) II=NL
    NDRAW(II)=1
    NLINE=1
    GO TO 280
710 CALL NEWPAG
    GO TO 50
C
C    RESTORE WINDOW (R)
C
720 IF(X0.GT.TXMIN.OR.NLINE.EQ.0) GO TO 730
    NLINE=0
    GO TO 280
730 IWIN=0
    GO TO 710
C
C    SHOW VALUE (S)
C
740 LCNT=LCNT-LDEL
    CALL MOVABS(0,LCNT)
    CALL ANMODE
    PRINT 750,X0,Y0
750 FORMAT(*X=*,G13.5,/,*Y=*,G13.5)
    LCNT=LCNT-LDEL
    GO TO 280
C
C    VALUE IN (V)
C
760 LCNT=LCNT-LDEL
    CALL ROTATE(0,LCNT,10,MSG5)
    LCNT=LCNT-LDEL
    CALL MOVABS(0,LCNT)
    CALL ANMODE
    CALL GETIN(2,VTEM)
    X0=VTEM(1)
    Y0=VTEM(2)
C
C    CHECK FOR N COMMAND VALUE INPUT SECTION.
C
    IF(IPLT.EQ.0) GO TO 600
    LCNT=LCNT-LDEL
    CALL ROTATE(0,LCNT,13,MSG6)
    CALL TINPUT(ICHR)
    GO TO 340
C
C    WINDOW (W)
C
770 CALL GETVAL(ICHR,X1,Y1)
780 AXMIN=AMIN1(X0,X1)
    AXMAX=AMAX1(X0,X1)
    AYMIN=AMIN1(Y0,Y1)
    AYMAX=AMAX1(Y0,Y1)
    IWIN=1
    IF(ICHR.EQ.122) GO TO 710
    IF((AYMAX-AYMIN)/DELY.GT.8.) GO TO 710
    LCNT=LCNT-LDEL
    CALL NOTATE(0,LCNT,29,MSG7)

```

```

TIGP0481
TIGP0482
TIGP0483
TIGP0484
TIGP0485
TIGP0486
TIGP0487
TIGP0488
TIGP0489
TIGP0490
TIGP0491
TIGP0492
TIGP0493
TIGP0494
TIGP0495
TIGP0496
TIGP0497
TIGP0498
TIGP0499
TIGP0500
TIGP0501
TIGP0502
TIGP0503
TIGP0504
TIGP0505
TIGP0506
TIGP0507
TIGP0508
TIGP0509
TIGP0510
TIGP0511
TIGP0512
TIGP0513
TIGP0514
TIGP0515
TIGP0516
TIGP0517
TIGP0518
TIGP0519
TIGP0520
TIGP0521
TIGP0522
TIGP0523
TIGP0524
TIGP0525
TIGP0526
TIGP0527
TIGP0528
TIGP0529
TIGP0530
TIGP0531
TIGP0532
TIGP0533
TIGP0534
TIGP0535
TIGP0536
TIGP0537
TIGP0538
TIGP0539
TIGP0540
TIGP0541
TIGP0542

```

```

      LCNT=LCNT-LDEL*2
      CALL MCVAABS(0,LCNT)
      CALL ANMCDE
      LCNT=LCNT-1
      CALL GETIN(4,VTEM)
      XO=VTEM(1)
      YO=VTEM(2)
      X1=VTEM(3)
      Y1=VTEM(4)
      DELY=1.E-99
      GO TO 780
C***** LOCATE TITLE
C*****
C*790 CALL SWCHAR(1)
      790 CALL INTITE(XO,YO)
**** CALL SWCHAR(0)
      GO TO 280
C***** DELETE ENTIRE LINE CODE
C*****
      800 IKILL=-1
      GO TO 390
      810 XO=X(KLINE)
      YO=Y(KLINE)
      ICHAR=4
      GO TO 390
      END
CDWIDE
      SUBROUTINE DWIDE(X,Y,IR)
      COMMON/TEKGPPR/LDEL,LCNT,MAXSR,LTV(17),EN(2),DELY,DELY
      DATA DS/8./
      IF(IR.EQ.-1)GO TO 200
      DX=(X-XE)/DELY
      DY=(Y-YE)/DELY
      ANG=1.5707963
      SL=1.E10
      IF(DX.EQ.0.) GO TO 120
      SL=DY/DX
      ANG=ATAN(SL)
      120 DXP=DS*SIN(ANG)
      DYP=DS*COS(ANG)
      IF(SL.NE.0.) DXP=-DXP
      IF((DX*DYP-DY*DXP).GE.0.)GO TO 190
      DXP=-DXP
      DYP=-DYP
      190 DXP=DXP*DELY
      DYP=DYP*DELY
      CALL DRAWA(XB+DXP,YB+DYP)
      CALL DRAWA(X+DXP,Y+DYP)
      CALL DRAWA(X,Y)
      200 XB=X
      YB=Y
      IR=0
      RETURN
      END
CGETVAL
      SUBROUTINE GETVAL(ICAR,XV,YV)
      COMMON/TEKGPPR/DUM(3),ICL,ITAB,ITAES,XS,YS,DUM2(20),MLINE
C      ICL=0 INITIALIZE TABLET
C      IC=0 NOT IN CONTINUOUS MCDE
C      ITAB=0 SCREEN CURSER
C*****

```

```

TIGP0543
TIGP0544
TIGP0545
TIGP0546
TIGP0547
TIGP0548
TIGP0549
TIGP0550
TIGP0551
TIGP0552
TIGP0553
TIGP0554
TIGP0555
TIGP0556
TIGP0557
TIGP0558
TIGP0559
TIGP0560
TIGP0561
TIGP0562
TIGP0563
TIGP0564
TIGP0565
TIGP0566
TIGP0567
TIGP0568

```

```

GETV0001
GETV0002
GETV0003
GETV0004
GETV0005
GETV0006
GETV0007

```



```

C**** GET VALUE AND CHARACTER FROM CROSS HAIRS OR TABLET
C****
C****
C**** CHECK FOR TABLE INPUTS
C****
  10 IF(ITAB.EQ.1)GO TO 20
    CALL VCURSR(ICCHAR,XV,YV)
C****
C**** CHECK TO SEE IF SCREEN COMMAND WAS TO ACTIVATE TABLET
C****
  IF(ICCHAR.NE.84)GO TO 30
  ITAB=1
  ICL=ITABS
C****
C**** SET TABLET LAST CHARACTER (ALSO USED AS A FLAG FOR INITIALIZATION)
C****
  20 ICHAR=ICL
  CALL TABVU(ICCHAR,XV,YV)
C****
C**** SAVE LAST TABLET CHARACTER COMMAND
C****
  ICL=ICCHAR
C*** IF IN TABLET MODE AND A NEW LINE COMMAND
C*** THEN TURN OFF COMMAND ...DEFAULT IT TO ADD AFTER
  IF(ICL.EQ.78)ICL=65
  NLINE=0
C
C   SET FLAG TO PLOT ALL LINES IN TABLET MODE
C
C****
C**** CHECK FOR TABLET HALT COMMAND
C****
  IF(ICL.NE.72)GO TO 30
C****
C**** TURN OFF TABLET AND SAVE LAST COMMAND
C****
  ITAB=0
  ITABS=ICL
  RETURN
  30 IF(ICCHAR.EQ.69)ITABS=0
  RETURN
END
CTABVU

```

GETV0008  
GETV0009  
GETV0010  
GETV0011  
GETV0012  
GETV0013  
GETV0014  
GETV0015  
GETV0016  
GETV0017  
GETV0018  
GETV0019  
GETV0020  
GETV0021  
GETV0022  
GETV0023  
GETV0024  
GETV0025  
GETV0026  
GETV0027  
GETV0028  
GETV0029  
  
GETV0030  
  
  
  
  
  
  
GETV0031  
GETV0032  
GETV0033  
GETV0034  
GETV0035  
GETV0036  
GETV0037  
GETV0038  
GETV0039  
GETV0040  
GETV0041  
GETV0042  
GETV0043

```

SUBROUTINE TABVU(ICCHAR,XV,YV)
COMMON/TEKGPPR/LDEL,LCNT,MAXSR,LTV(5)
1 ,LS,MX1,MY1,MX2,MY2,XB,YB,FACX,FACY,ANG,MXB,MYB
DIMENSION MSG1(54),MSG2(43),MSG3(43),MSG4(18),ICONV(2,10),
1 IRETN(2,10),XTEM(2)
DATA ((ICONV(I,J),J=1,10),I=1,2)/65,66,67,68,69,71,72,
1 105,78,80,82,83,86,87,
1 32,32,32,32,32,32/
DATA ((IRETN(I,J),J=1,10),I=1,2)/-0,-0,-0,0,1,1,1,
1 1,0,1,1,0,1,0,
1 1,1,1,1,1,1,1/
DATA MSG1/ 83,113,117, 97,114,101, 32,109,101,110,
1 117, 32,119,105,116,104, 32,116, 97, 98,
1 108,101,116, 32, 97,110,100, 32,116,111,
1 117, 99,104, 32,117,112,112,101,114, 32,
1 108,101,102,116, 32,109,101,110,117, 32,
1 100,111,116, 46/
DATA MSG2/ 84,111,117, 99,104, 32, 97,120,105,115,

```

TAEV0001  
  
TABV0002  
TABV0003  
TABV0004  
TABV0005  
TABV0006  
TABV0007  
TABV0008  
TABV0009  
TABV0010  
TABV0011  
TABV0012  
TABV0013  
TABV0014  
TABV0015  
TABV0016  
TABV0017  
TABV0018  
TABV0019

```

1      32, 111, 114, 105, 103, 105, 110, 32, 97, 110,
1      100, 32, 101, 110, 116, 101, 114, 32, 118, 97,
1      108, 117, 101, 115, 32, 88, 32, 97, 110, 100,
1      32, 89, 46/
DATA MSG3/ 84, 111, 117, 99, 104, 32, 32, 32, 97, 120,
1      105, 115, 32, 97, 116, 32, 109, 97, 120, 32,
1      108, 101, 110, 102, 116, 104, 32, 97, 110, 100,
1      32, 101, 110, 116, 101, 114, 32, 118, 97, 108,
1      117, 101, 46/
DATA MSG4/ 76, 97, 115, 116, 32, 99, 111, 109, 109, 97,
1      110, 100, 32, 119, 97, 115, 32, 32/
IF(ICHAR.NE.O)GO TO 30
C      TAELET HAS NOT BEEN SET CHECK IT
      LS=100
      CALL TAEINT(1,C,O)
      CALL NEWPAG
      LCNT=3120-LDEL
C      GET MENU POSITION
      CALL NCTATE(O,LCNT,54,MSG1)
      CALL BELL
      CALL ONEPNT(MX1,MY1)
      MX2=MX1+1000
      MY2=MY1-200
      GO TO 20
10 LCNT=3120
      CALL NEWPAG
20 LCNT=LCNT-LDEL
C      GET COORDINATE INTERSECTION
      CALL NCTATE(O,LCNT,43,MSG2)
      CALL BELL
      CALL ONEPNT(MXB,MYB)
      LCNT=LCNT-LDEL
      CALL MOVABS(O,LCNT)
      CALL ANMCDE
      CALL GETIN(2,XTEM)
      XB=XTEM(1)
      YB=XTEM(2)
      LCNT=LCNT-LDEL
      MSG3(7)=88
C      GET X AXIS POSITION MAX
      CALL NCTATE(O,LCNT,43,MSG3)
      CALL BELL
      CALL ONEPNT(MX1,NX1)
      LCNT=LCNT-LDEL
      CALL MOVABS(O,LCNT)
      CALL ANMCDE
C      GET VALUE AT POSITION
      CALL GETIN(1,XM)
      DX=NXM-MXB
      DY=NYM-MYB
C      COMPUTE ANGLE CORRECTION
      ANG=ATAN2(DY,DX)
      LCNT=LCNT-LDEL
      MSG3(7)=89
C      GET Y AXIS POSITION MAX
      CALL NCTATE(O,LCNT,43,MSG3)
      CALL BELL
      CALL ONEPNT(MYM,NYM)
      LCNT=LCNT-LDEL
      CALL MOVABS(O,LCNT)
      CALL ANMCDE
C      GET VALUE AT POSITION

```

```

TABV0020
TABV0021
TABV0022
TABV0023
TABV0024
TABV0025
TABV0026
TABV0027
TABV0028
TABV0029
TABV0030
TABV0031
TABV0032
TABV0033
TABV0034
TABV0035
TABV0036
TABV0037
TABV0038
TABV0039
TABV0040
TABV0041
TABV0042
TABV0043
TABV0044
TABV0045
TABV0046
TABV0047
TABV0048
TABV0049
TABV0050
TABV0051
TABV0052
TABV0053
TABV0054
TABV0055
TABV0056
TABV0057
TABV0058
TABV0059
TABV0060
TABV0061
TABV0062
TABV0063
TABV0064
TABV0065
TABV0066
TABV0067
TABV0068
TABV0069
TABV0070
TABV0071
TABV0072
TABV0073
TABV0074
TABV0075
TABV0076
TABV0077
TABV0078
TABV0079
TABV0080
TABV0081

```

```

CALL GETIN(1,YM)
DY=NYM-MYB
COSA=COS(ANG)
C SET UP COMMON FACTORS FOR ANGLE CORRECTIONS
FACX=(XM-XB)*COSA/DX
FACY=(YM-YE)*COSA/DY
INIT=1
XV=XM
YV=YM
ICAR=27
C RETURN PLOT COMMAND
RETURN
C CHECK FOR TABLET INITIALIZED
30 IF(INIT.NE.1)GO TO 40
ICAR=122
XV=XB
YV=YB
INIT=0
RETURN
40 CALL BELL
CALL ONEPNT(IX,IY)
C CHECK TO SEE IF POINT SENT IS A MENU COMMAND
IF(IX.GT.MX2.OR.IX.LT.MX1)GO TO 50
IF(IY.GT.MY1.OR.IY.LT.MY2)GO TO 50
IC=(IX-MX1)/LS+1
IR=(MY1-IY)/LS+1
C CONVERT ROW AND COLUMN POSITION TO COMMAND CHARACTER
ICAR=ICONV(IR,IC)
IF(ICAR.EQ.32) RETURN
MSG4(12)=ICAR
LCNT=LCNT-LDEL
C LAST MESSAGE COMMAND
CALL NOTATE(0,LCNT,18,MSG4)
IF(ICAR.EQ.105)GO TO 10
IF(IRETN(IR,IC).EQ.1)RETURN
GO TO 40
C CONVERT TABLET UNITS TO VIRTUAL UNITS WITH ANGLE CORRECTION
50 DX=IX-MXB
DY=IY-MYB
IF(DX.EC.0.)DX=1.E-20
R=SQRT(DX*DX+DY*DY)
ANGR=ATAN2(DY,DX)-ANG
XV=R*FACX*COS(ANGR)+XB
YV=R*FACY*SIN(ANGR)+YB
RETURN
END
CDRAWIT
SUERCUTINE DRAWIT(NL,NPTA,X,Y,ISUB,MTT,LTT)
COMMON /SHAD/KSHADE(2,168)
COMMON/TKTRNX/ITEKC(60)
COMMON/TEKGPPR/DUM(20),EN(2),DEL(2),BEG(2),RDX2,RDY2,NLINE,
1 NDRAW(50)
EQUIVALENCE (IOFF,ITEKC(30))
DIMENSION QSY(626),CSX(626),NPTA(1),X(1),Y(1),ISUB(1)
DIMENSION MCCN(16),IDLN(10),MTT(1),LTT(1)
DATA MCON/2,1,2,3,4,5,0,0,0,0,0,-1,-2,-3,-4,-5/
DATA IDLN/
+0,56,
+776,
+77616,
+77777777616,
+77761616161,

```

TABV0082  
 TABV0083  
 TABV0084  
 TABV0085  
 TABV0086  
 TABV0087  
 TABV0088  
 TABV0089  
 TABV0090  
 TABV0091  
 TABV0092  
 TABV0093  
 TABV0094  
 TABV0095  
 TABV0096  
 TABV0097  
 TABV0098  
 TABV0099  
 TABV0100  
 TABV0101  
 TABV0102  
 TABV0103  
 TABV0104  
 TABV0105  
 TABV0106  
 TABV0107  
 TABV0108  
 TABV0109  
 TABV0110  
 TABV0111  
 TABV0112  
 TABV0113  
 TABV0114  
 TABV0115  
 TABV0116  
 TABV0117  
 TABV0118  
 TABV0119  
 TABV0120  
 TABV0121  
 TABV0122  
 TABV0123  
 TABV0124  
 TABV0125  
 TABV0126  
 TABV0127  
 DRAW0001

DRAW0003  
 DRAW0004

DRAW0007  
 DRAW0006

+777777776.  
 +777777616161.  
 +776655443322.  
 +18/

C		DRAW0009
C		DRAW0010
C	0 SYMBOLS 1 LINE 2 SFLINE WRT X 3 SPLINE WRT Y 4 ARC FIT 5 CLOSED	DRAW0012
	DO 1 I=1,166	
	KSHADE(2,I)=0	
	1 KSHADE(1,I)=0	
	IT=64	
	K=1	DRAW0013
	NSUM=1	DRAW0014
	NC=0	DRAW0015
	DO 290 I=1,NL	
	IR=-1	DRAW0016
	KIND=LTT(I)+1	
	LDASH=IDLN(KIND)	
	KIND=MTT(I)+1	
	IWID=0	
	2 IF(KIND.LT.20)GO TO 5	
	IWID=IWID+1	
	KIND=KIND-10	
	GO TO 2	
	5 ISYM=MCON(KIND)	
	ITYP=IAES(ISYM)	
	IF(ITYP.GT.1) GO TO 40	DRAW0008
	NEND=NSUM+NPTA(I)-1	DRAW0017
	IT=IT+1	DRAW0018
	IF(IT.GT.90)IT=65	DRAW0019
	DO 20 J=NSUM,NEND	DRAW0020
	IF(NLINE.EC.0) GO TO 10	DRAW0021
	IF(NDRAW(I).EC.0) GO TO 20	DRAW0022
	10 XP=X(K)	DRAW0023
	YP=Y(K)	DRAW0024
	IF(J.EC.NSUM) CALL MCVEA(XP,YP)	DRAW0025
	IF(LDASH.EC.0)GO TO 16	
	CALL DASHA(XP,YP,LDASH)	
	GO TO 15	
	16 IF(IWID.NE.2)GO TO 17	
	CALL SHADE(XP,YP,IR)	
	GO TO 15	
	17 CALL DRAWA(XP,YP)	
	IF(IWID.EC.1)CALL DWIDE(XP,YP,IR)	
	15 IF(ISYM.LT.0) GO TO 20	DRAW0027
	CALL MCVEA(XP,YP)	DRAW0028
	IF(IOFF.EC.0) CALL ANCHO(IT)	DRAW0029
	CALL MCVEA(XP,YP)	DRAW00
	20 K=ISUB(K)	DRAW0030
	30 NSUM=NEND+1	DRAW0031
	GO TO 290	DRAW0033
C		DRAW0034
C	PLOT WITH SFLINE	DRAW0035
C		DRAW0036
	40 NS=NC	DRAW0042
	NPT=NPTA(I)	DRAW0043
	NC=NC+NPT	DRAW0044
	IT=IT+1	DRAW0045
	IF(IT.GT.90)IT=65	DRAW0046
	IF(NLINE.EC.0) GO TO 60	DRAW0047
	IF(NDRAW(I).NE.0) GO TO 60	DRAW0048

```

C      LOCATE POINTER OF NEXT LINE
      DO 50 L=1,NPT
50    K=ISUB(K)
      GO TO 290
60    JFIT=2
      YO=Y(K)
      K1=ISUB(K)
      IF(ITYP.GT.2) GO TO 80
      XO=X(K)
C      CHECK X DATA FOR ASCENDING ORDER
      DO 70 L=2,NPT
      X1=X(K1)
      IF(X1.LE.XO) GO TO 110
      K1=ISUB(K1)
70    XO=X1
      GO TO 210
80    IF(ITYP.GT.3) GO TO 100
C      CHECK Y DATA FOR ASCENDING ORDER
      DO 90 L=2,NPT
      Y1=Y(K1)
      IF(Y1.LE.YO) GO TO 110
      K1=ISUB(K1)
90    YO=Y1
      GO TO 210
100   JFIT=ITYP-2
110   NCIR=0
      IF(JFIT.EQ.3) NCIR=-NPT/2-1
      IF(NCIR.LT.-5) NCIR=-5
      MPT=NPT-2*NCIR
      QSY(1)=MPT
      QSX(1)=MPT
      S=0.
      KA=NS
      KO=KA
      KE=KO+NPT
      KSAVE=K
      KA=KA+NCIR
      DO 160 M=1,MPT
      M1=M+1
      KA=KA+1
      IF(KA.GT.KO) GO TO 130
      NDO=NPT+NCIR
      DO 120 II=1,NDO
120   K=ISUB(K)
      KA=KA+NPT,
      GO TO 140
130   IF(KA.NE.(KE+1))GO TO 140
      JSAVE=K
      K=KSAVE
      KA=KA-NPT
140   CONTINUE
      L=M1+NPT
      YYYP=Y(K)
      XXXP=X(K)
      K=ISUB(K)
      IF(K.EQ.1) GO TO 150
      DS=SCRT(RDX2*(XXXP-XO)**2+RDY2*(YYYP-YO)**2)
      S=S+DS
150   XO=XXXP
      YO=YYYP
      QSX(M1)=S
      QSY(M1)=S
      QSX(L)=XXXP

```

```

DRAW0049
DRAW0050
DRAW0051
DRAW0052
DRAW0053
DRAW0054
DRAW0055
DRAW0056
DRAW0057
DRAW0058
DRAW0059
DRAW0060
DRAW0061
DRAW0062
DRAW0063
DRAW0064
DRAW0065
DRAW0066
DRAW0067
DRAW0068
DRAW0069
DRAW0070
DRAW0071
DRAW0072
DRAW0073

DRAW0075

DRAW0076
DRAW0077
DRAW0078
DRAW0079
DRAW0080
DRAW0081
DRAW0082
DRAW0083
DRAW0084
DRAW0085
DRAW0086
DRAW0087
DRAW0088
DRAW0089
DRAW0090
DRAW0091
DRAW0092
DRAW0093
DRAW0094
DRAW0095
DRAW0096
DRAW0097
DRAW0098
DRAW0099
DRAW0100
DRAW0101
DRAW0102
DRAW0103
DRAW0104
DRAW0105
DRAW0106
DRAW0107
DRAW0108
DRAW0109
DRAW0110

```

160	CSY(L)=YYYY	
	KA=KO+NPT	
	CSX(L+1)=0.	
	CSY(L+1)=0.	
	CSY(L+2)=1.	
	CSY(L+2)=1.	
	XG=CSX(MPT+2-NCIR)	
	YO=CSY(MPT+2-NCIR)	
	IF(ISYM.LE.0) GO TO 169	
	CALL MOVEA(XO,YO)	
	IF(IOFF.EQ.0) CALL ANCHO(IT)	
169	CALL MOVEA(XC,YO)	
	IF(IWID.EQ.1)CALL DWIDE(XC,YO,IR)	
	IF(IWID.EQ.2)CALL SHADE(XO,YO,IR)	
	SCK=QSX(3-NCIR)	
	S=QSX(2-NCIR)	
	IF(NPT.LE.1) GO TO 290	
	DC=40.	
	DS=40.	
	NCK=2	
170	S=S+DS	
	XP=SPLNQ1(1,CSX,S)	
	YP=SPLNQ1(1,CSY,S)	
	DCK=SCRT(RDX2*(XC-XP)**2+RDY2*(YO-YP)**2)	
	DS= DC*DS/DCK	
180	IF(S.LT.SCK) GO TO 200	
	NSYM=MPT+1+NCK-NCIR	
	XS=QSX(NSYM)	
	YS=QSY(NSYM)	
	IF(LDASH.EQ.0)GO TO 185	
	CALL DASHA(XP,YP,LDASH)	
	GO TO 186	
185	IF(IWID.NE.2)GO TO 187	
	CALL SHADE(XS,YS,IR)	
	GO TO 186	
187	CALL DRAWA(XS,YS)	
	IF(IWID.EQ.1)CALL DWIDE(XS,YS,IR)	
186	IF(ISYM.LE.0)GO TO 190	
	CALL MOVEA(XS,YS)	
	IF(IOFF.EQ.0) CALL ANCHO(IT)	
	CALL MOVEA(XS,YS)	
190	NCK=NCK+1	
	SCK=QSX(NCK+1-NCIR)	
	IF(NCK.LE.NPT+JFIT-2) GO TO 180	
	IF(JFIT.EQ.3) K=JSAVE	
	GO TO 290	
200	IF(LDASH.EQ.0)GO TO 205	
	CALL DASHA(XP,YP,LDASH)	
	GO TO 206	
205	IF(IWID.NE.2) GO TO 207	
	CALL SHADE(XP,YP,IR)	
	GO TO 206	
207	CALL DRAWA(XP,YP)	
	IF(IWID.EQ.1)CALL DWIDE(XP,YP,IR)	
206	XO=XP	
	YO=YP	
	GO TO 170	
210	QSX(1)=NPT	
	DO 240 M=1,NPT	
	N=M+1	
	KA=NS+M	
	L=N+NPT	
	XP=X(K)	

DRAW0111
DRAW0112
DRAW0113
DRAW0114
DRAW0115
DRAW0116
DRAW0117
DRAW0118
DRAW0119
DRAW0120
DRAW0121
DRAW0122
DRAW0123
DRAW0124
DRAW0125
DRAW0126
DRAW0127
DRAW0128
DRAW0129
DRAW0130
DRAW0131
DRAW0132
DRAW0133
DRAW0134
DRAW0135
DRAW0136
DRAW0137
DRAW0139
DRAW0140
DRAW0141
DRAW0142
DRAW0143
DRAW0144
DRAW0145
DRAW0146
DRAW0148
DRAW0149
DRAW0150
DRAW0151
DRAW0152
DRAW0153
DRAW0154
DRAW0155
DRAW0156

```

      YP=Y(K)
      IF(ISYM.LE.0) GO TO 220
      CALL MCVEA(XP,YP)
      IF(IOFF.EQ.0) CALL ANCHO(IT)
220  IF(ITYP.NE.3) GO TO 230
      QSX(N)=YP
      QSX(L)=XP
      GO TO 240
230  QSX(N)=XP
      QSX(L)=YP
240  K=ISUB(K)
      QSX(L+1)=0.
      QSX(L+2)=1.
      XEN=QSX(NPT+1)
      XIN=QSX(2)
      IFITP=ITYP-1
      BCK=BEG(IFITP)
      ECK=EN(IFITP)
      DELT=DEL(IFITP)*30.
      IF(XIN.LT.ECK) XIN=BCK
      IF(XEN.GT.ECK) XEN=ECK
      KILL=0
      DO 280 M=1,200
      XI=XIN+DELT*(M-1)
      IF(XI.LT.XEN) GO TO 250
      KILL=1
      XI=XEN
250  YI=SPLNC1(1, QSX, XI)
      IF(ITYP.EQ.3) GO TO 260
      XP=XI
      YP=YI
      GO TO 270
260  XP=YI
      YP=XI
270  IF(M.EQ.1) CALL MOVEA(XP,YP)
      IF(LDASH.EQ.0) GO TO 275
      CALL DASHA(XP,YP,LDASH)
      GO TO 276
275  IF(IWID.NE.2) GO TO 277
      CALL SHADE(XP,YP,IR)
      GO TO 276
277  CALL DRAWA(XP,YP)
      IF(IWID.EQ.1) CALL DWIDE(XP,YP,IR)
276  IF(NPT.EQ.1) GO TO 290
      IF(KILL.EQ.1) GO TO 290
280  CONTINUE
290  CONTINUE
300  RETURN
      END
CSHADE
      SUBROUTINE SHADE(XO,YO,IR)
      COMMON/TEKGPPR/LDEL,LCNT,MAXSR,LTV(17),EN(2),DEL(2),BEG(2)
      COMMON /SHAD/KSHADE(2,168)
      IXTOS(X)=((X-BEG(1))/DEL(1)+600)
      IYTOS(Y)=((Y-BEG(2))/DEL(2)+300)
      X=XO
      Y=YO
      IF(IR.EQ.-1) GO TO 200
      IF(XB.LT.X) GO TO 10
      X2=XB
      Y2=YP
      XB=X

```

```

DRAW0157
DRAW0159
DRAW0160
DRAW0161
DRAW0162
DRAW0163
DRAW0164
DRAW0165
DRAW0166
DRAW0167
DRAW0168
DRAW0169
DRAW0170
DRAW0171
DRAW0172
DRAW0173
DRAW0174
DRAW0175
DRAW0176
DRAW0177
DRAW0178
DRAW0179
DRAW0180
DRAW0181
DRAW0182
DRAW0183
DRAW0184
DRAW0185
DRAW0186
DRAW0187
DRAW0188
DRAW0189
DRAW0190
DRAW0191

```

```

DRAW0194
DRAW0195
DRAW0196
DRAW0197
DRAW0198

```

```

TIGP00
TIGP00

```

[illegible]



```

59
IF(IPX.GT.IMAXX)GO TO 100
IF(IY1.EQ.IY2)GO TO 50
S=(IX2-IX1)/FLOAT(IY2-IY1)
B=FLCAT(IX1)-S*IY1
IY1=IFIX((FLOAT(IPX)-B)/S)
50 LL=L
IF(IY1.GT.IPEND)GO TO 100
IF(K.EQ.0)GO TO 95
DO 90 M1=1,K
IF(IY1.GT.IPP(M1))GO TO 90
IPS1=IPP(M1)
IPP(M1)=IY1
IY1=IPS1
IPS1=ITN(M1)
ITN(M1)=LL
LL=IPS1
90 CONTINUE
95 K=K+1
ITN(K)=LL
IPP(K)=IY1
100 J=I
101 CONTINUE
NC=0
K=K+1
IPP(K)=IPEND
ITN(K)=0
IPO=300
ISUB=0
DO 200 I=1,K
L=ITN(I)
IPY=IPP(I)
IF(NC.EQ.0)GO TO 160
DO 120 J=1,NC
IF(IDN(J).EQ.L) GO TO 130
120 CONTINUE
NC=NC+1
IF(NC.GT.10)STOP "ERROR IN NC SHADEIT"
IDN(NC)=L
GO TO 140
130 IDN(J)=IDN(NC)
NC=NC-1
140 IDX=(IPY/20)*20+20-IPY
IPO=IPY
GO TO 200
160 IF(IPY.LT.IPO) GO TO 200
165 IF(IPY.LT.(IPO+IDX)) GO TO 180
IPO=IPO+IDX
IDX=20
ISUB=FLOAT(IPO-300)/20.
IW=1
IF(ISUB.GT.60)IW=2
IWS=KSHADE(IW,K)
ICP=SHIFT(1,(60*IW-ISUB+1))
IF((ICP.AND.IWS).EQ.0)GO TO 165
CALL PNTABS(IPX,IPC)
GO TO 165
180 CONTINUE
IPO=IPY
NC=NC+1
IDN(NC)=L
200 CONTINUE
1000 CONTINUE
END

```

```

CSPLNC1
  FUNCTION SPLNC1 (NLOC,X,XINDEP)
    COMMON/SPLCO/DX,CM
C*** LOCAL CUBIC FIT 8/9/77 M.J. CADDY
    DIMENSION X(1),CM(3)
    EQUIVALENCE (CM(1),T3),(CM(2),Q2),(CM(3),Q3)
    XIN=XINDEP
    NS=NLOC
    NOPTS=X(NS)
    ID=NS+NOPTS
    NSP1=NS+1
    NSP2=NS+2
    IF(NOPTS.LE.1) GO TO 130
    IF(NOPTS.GT.2) GO TO 10
    N=ID+NOPTS
    T3=(X(N)-X(N-1))/(X(ID)-X(ID-1))
    H=ID
    NTRAP=1
    GO TO 280
  10 NS2=NOPTS*2+NSP1
    L=X(NS2)
    LSC=NS2+1
    IQMODE=X(LSC)
    K=L+NS
    NL=NSP1
    NH=ID
    NTRAP=-1
C*** BINARY SEARCH FOR INTERVAL
    IF(XIN-X(ID))30,140,20
  20 NTRAP=0
    GO TO 150
  30 IF(XIN-X(NSP1))40,40,60
  40 NTRAP=1
  50 K=NSP2
    GO TO 160
  60 IF(L)120,120,70
  70 IF(XIN-X(K))80,100,100
  80 NH=K
    K=K-1
  90 IF(XIN-X(K))110,100,100
  100 NL=K
    GO TO 120
  110 NH=K
  120 K=(NH-NL)/2+NL
    IF(K-NL)90,140,90
  130 YOUT=X(NSP2)
    GO TO 320
  140 LFAST=L-NH+NS
    X(NS2)=NH-NS
  150 K=NH
  160 M=K
    N=M+NOPTS
    Y3=X(N-1)
    X3=X(M-1)
C*** CHECK FOR FAST MODE AND EXTRAPOLATION
    IF(NTRAP.GE.0) GO TO 180
    IF(IQMODE*L.EQ.0.OR.LFAST.NE.0) GO TO 180
    DO 170 I=1,3
  170 CM(I)=X(LSC+I)
    GO TO 310
  180 Y4=X(N)
    X4=X(K)

```

```

SPLN0001
SPLN0002
SPLN0003
SPLN0004
SPLN0005
SPLN0006
SPLN0007
SPLN0008
SPLN0009
SPLN0010
SPLN0011
SPLN0012
SPLN0013
SPLN0014
SPLN0015
SPLN0016
SPLN0017
SPLN0018
SPLN0019
SPLN0020
SPLN0021
SPLN0022
SPLN0023
SPLN0024
SPLN0025
SPLN0026
SPLN0027
SPLN0028
SPLN0029
SPLN0030
SPLN0031
SPLN0032
SPLN0033
SPLN0034
SPLN0035
SPLN0036
SPLN0037
SPLN0038
SPLN0039
SPLN0040
SPLN0041
SPLN0042
SPLN0043
SPLN0044
SPLN0045
SPLN0046
SPLN0047
SPLN0048
SPLN0049
SPLN0050
SPLN0051
SPLN0052
SPLN0053
SPLN0054
SPLN0055
SPLN0056
SPLN0057
SPLN0058
SPLN0059
SPLN0060

```

```

      A3=X4-X3
      S3=(Y4-Y3)/A3
      IF(M.EQ.NSP2) GO TO 190
      X2=X(M-2)
      Y2=X(N-2)
      S2=(Y3-Y2)/(X3-X2)
      IF(M.EQ.ID) GO TO 200
190  X5=X(M+1)
      Y5=X(N+1)
      S4=(Y5-Y4)/(X5-X4)
      IF(M.EQ.NSP2) S2=S3+S3-S4
      GO TO 210
200  S4=S3+S3-S2
210  IF(M.LE.(NSP2+1)) GO TO 220
      S1=(Y2-X(N-3))/(X2-X(M-3))
      GO TO 230
220  S1=S2+S2-S3
230  IF(M.GE.(ID-1)) GO TO 240
      S5=(X(N+2)-Y5)/(X(M+2)-X5)
      GO TO 250
240  S5=S4+S4-S3
250  W2=ABS(S4-S3)
      W3=ABS(S2-S1)
      SW=W2+W3
      IF(SW.NE.0.0) GO TO 260
      W2=0.5
      W3=0.5
      SW=1.0
260  T3=(W2*S2+W3*S3)/SW
      W3=ABS(S5-S4)
      W4=ABS(S3-S2)
      SW=W3+W4
      IF(SW.NE.0.0) GO TO 270
      W3=0.5
      W4=0.5
      SW=1.0
270  T4=(W3*S3+W4*S4)/SW
      IF(NTRAP.LT.0) GO TO 290
      IF(NTRAP.EQ.0) T3=T4
280  IX=M-NTRAP
C*** FAST EXIT FOR 2 POINTS AND LINEAR EXTRAPOLATION
      YOUT=X(IX+NOPTS)+(XIN-X(IX))*T3
      GO TO 320
290  Q2=(2.0*(S3-T3)+S3-T4)/A3
      Q3=(-S3-S3+T3+T4)/(A3*A3)
      IF(IQMODE*LFast.EQ.0) GO TO 310
      DO 300 I=1,3
300  X(LSC+I)=QH(I)
310  DX=XIN-X3
      YOUT=Y3+DX*(T3+DX*(Q2+DX*Q3))
320  SPLNQ1=YOUT
      RETURN
      END
CPTITE
      SUBROUTINE PTITE(IXI,IYI,NTL,LATL,IC,NM,AT,IPX,IPY)
      DIMENSION LATL(1),IP(136),IPX(4),IPY(4)
C      NTL =NUMBER OF 10 CHARACTER WORDS
C      NM MAX CHARACTERS PER LINE
C      AT ANGLE OF TITLE
C      IX SCREEN CENTER
C      IY SCREEN CENTER

```

```

SPLN0061
SPLN0062
SPLN0063
SPLN0064
SPLN0065
SPLN0066
SPLN0067
SPLN0068
SPLN0069
SPLN0070
SPLN0071
SPLN0072
SPLN0073
SPLN0074
SPLN0075
SPLN0076
SPLN0077
SPLN0078
SPLN0079
SPLN0080
SPLN0081
SPLN0082
SPLN0083
SPLN0084
SPLN0085
SPLN0086
SPLN0087
SPLN0088
SPLN0089
SPLN0090
SPLN0091
SPLN0092
SPLN0093
SPLN0094
SPLN0095
SPLN0096
SPLN0097
SPLN0098
SPLN0099
SPLN0100
SPLN0101
SPLN0102
SPLN0103
SPLN0104
SPLN0105
SPLN0106
SPLN0107
SPLN0108
SPLN0109
SPLN0110
SPLN0111
SPLN0112
SPLN0113

```

```

      65
C      IC =1 NOT CENTERED TITLE
      IF(NTL.LE.0) RETURN
      NC=10*NTL
      CO=1.
      SI=0.
C      GET CHARACTER SIZE
      CALL CSIZE(IHORZ,IVERT)
C      CONVERT LABEL TO ADE
      NR=NC
      IF(NC.GT.70)NR=70
      CALL KAM2AS(NR,LABTL,IP)
      IF(NR.EQ.NC)GO TO 5
      NR=NC-70
      CALL KAM2AS(NR,LABTL(8),IP(71))
5     IX=IXI
      IY=IYI
      IX1=IX
      IY1=IY
      ITL1=0
      NCM=0
      NL=0
      NBLK=0
      CALL LINROT(AT)
      ANG=AT*.01745329252
      CO=COS(ANG)
      SI=SIN(ANG)
      DO 70 K=1,NC
C      CHECK FOR LEADING BLANKS
      IF(IP(K).NE.32) GO TO 10
      IF(ITL1.EQ.0) GO TO 70
      NBLK=NBLK+1
C      CHECK FOR 3 BLANKS TO TERMINATE LINE
      IF(NBLK.NE.3) GO TO 20
      ITL1=ITL1-2
      GO TO 50
10     NBLK=0
C      CHECK FOR MAX LINE LENGTH EXCEEDED
20     IF(ITL1.LT.NM) GO TO 30
      IF(IP(K).EQ.32) GO TO 50
30     ITL1=ITL1+1
      IP(ITL1)=IP(K)
      IF(K.LT.NC) GO TO 70
40     ITL1=ITL1-NBLK
C      CHECK FOR VERTICAL OR HORIZ LABEL
50     IF(ITL1.GT.NCM)NCM=ITL1
      NL=NL+1
      IX1=IX-IHORZ*ITL1*.5*CO*(1-IC)
      IY1=IY1-IHORZ*ITL1*.5*SI*(1-IC)
      CALL NOTATE(IX1,IY1,ITL1,IP)
      IX=IX+IVERT*1.1*SI
      IY1=IY1-IVERT*1.1*CO
      ITL1=0
70     CONTINUE
      XH=.5*IHORZ*NCM+.5*IHORZ
      YH=1.0*IVERT
      XHH=.5*IHORZ*NCM*IC
      IXI=IXI+CO*XHH
      IYI=IYI+SI*XHH
      IPX(1)=IXI-XH*CO-YH*SI
      IPY(1)=IYI-XH*SI+YH*CO
      IPX(2)=IXI+XH*CO-YH*SI
      IPY(2)=IYI+XH*SI+YH*CO

```

67

```

YH=(1.1*IVERT*(NL-1)+.3*IVERT)
IPY(4)=IYI-XH*SI-YH*CO
IPX(4)=IXI-XH*CO+YH*SI
IPX(3)=IXI+XH*CO+YH*SI
IPY(3)=IYI+XH*SI-YH*CO
END

```

CPGRIDX

```

SUBROUTINE PGRIDX(ITX,ITY,NTIT,NXU,NYU,IDX,ISKIP)
DIMENSION ITX(4,50),ITY(4,50),IPP(100),ITN(100),IDN(10)
IPEND=3960
DEL=2400./NYU
NMAX=NYU+1
DO 1000 N=1,NMAX,ISKIP
IPY=300.+DEL*(NMAX-N)
K=0
DO 101 L=1,NTIT
J=4
DO 100 I=1,4
IX1=ITX(J,L)
IX2=ITX(I,L)
IY1=ITY(J,L)
IY2=ITY(I,L)
IMINY=MIN0(IY1,IY2)
IF(IPY.LE.IMINY) GO TO 100
IMAXY=MAX0(IY1,IY2)
IF(IPY.GT.IMAXY) GO TO 100
IF(IX1.EQ.IX2) GO TO 50
S=(IY2-IY1)/FLOAT(IX2-IX1)
E=FLOAT(IY1)-S*IX1
IX1=IFIX((FLOAT(IPY)-B)/S)
50 LL=L
IF(IX1.GT.IPEND) GO TO 100
IF(K.EQ.0) GO TO 95
DO 90 M1=1,K
IF(IX1.GT.IPP(M1)) GO TO 90
IPS1=IPP(M1)
IPP(M1)=IX1
IX1=IPS1
IPS1=ITN(M1)
ITN(M1)=LL
LL=IPS1
90 CONTINUE
95 K=K+1
ITN(K)=LL
IPP(K)=IX1
100 J=I
101 CONTINUE
NC=0
K=K+1
IPP(K)=IPEND
ITN(K)=0
IPO=600
CALL MOVABS(IPO,IPY)
DO 200 I=1,K
L=ITN(I)
IPX=IPP(I)
IF(NC.EQ.0) GO TO 160
DO 120 J=1,NC
IF(IDN(J).EQ.L) GO TO 130
120 CONTINUE
NC=NC+1
IDN(NC)=L

```

```

      GO TO 140
130 IDN(J)=IDN(NC)
      NC=NC-1
140 CALL MOVABS(IPX,IPY)
      IPO=IPX
      GO TO 200
160 IF(IPX.LT.IPO) GO TO 200
165 IF(IPX.LT.(IPO+IDX)) GO TO 180
      IPO=IPO+IDX
      CALL DRWABS(IPO,IPY)
      GO TO 165
180 CALL DRWABS(IPX,IPY)
      IPO=IPX
      NC=NC+1
      IDN(NC)=L
200 CONTINUE
1000 CONTINUE
      END
CPGRIDY
      SUBROUTINE PGRIDY(ITX,ITY,NTIT,NXU,NYU,IDX,ISKIP)
      DIMENSION ITX(4,50),ITY(4,50),IPP(100),ITN(100),IDN(10)
      IPEND=2700
      DEL=3360./NXU
      NMAX=NXU+1
      DO 1000 N=1,NMAX,ISKIP
      IPX=600.+DEL*(NMAX-N)
      K=0
      DO 101 L=1,NTIT
      J=4
      DO 100 I=1,4
      IX1=ITX(J,L)
      IX2=ITX(I,L)
      IY1=ITY(J,L)
      IY2=ITY(I,L)
      IMINX=MINO(IX1,IX2)
      IF(IPX.LE.IMINX) GO TO 100
      IMAXX=MAXO(IX1,IX2)
      IF(IPX.GT.IMAXX) GO TO 100
      IF(IY1.EQ.IY2) GO TO 50
      S=(IX2-IX1)/FLOAT(IY2-IY1)
      B=FLOAT(IX1)-S*IY1
      IY1=IFIX((FLOAT(IPX)-B)/S)
50 LL=L
      IF(IY1.GT.IPEND) GO TO 100
      IF(K.EQ.0) GO TO 95
      DO 90 M=1,K
      IF(IY1.GT.IPP(M)) GO TO 90
      IPS1=IPP(M)
      IPP(M)=IY1
      IY1=IPS1
      IPS1=ITN(M)
      ITN(M)=LL
      LL=IPS1
90 CONTINUE
95 K=K+1
      ITN(K)=LL
      IPP(K)=IY1
100 J=J+1
101 CONTINUE
      NC=0
      K=K+1
      IPP(K)=IPEND

```

```

      ITN(K)=0
      IPO=300
      CALL MCVABS(IPX,IPO)
      DO 200 I=1,K
      L=ITN(I)
      IPY=IPP(I)
      IF(NC.EQ.0)GO TO 160
      DO 120 J=1,NC
      IF(IDN(J).EQ.L) GO TO 130
120  CONTINUE
      NC=NC+1
      IDN(NC)=L
      GO TO 140
130  IDN(J)=IDN(NC)
      NC=NC-1
140  CALL MCVABS(IPX,IPY)
      IPO=IPY
      GO TO 200
160  IF(IPY.LT.IPO) GO TO 200
165  IF(IPY.LT.(IPO-IDX)) GO TO 180
      IPO=IPO-IDX
      CALL DRWAES(IPX,IPO)
      GO TO 165
180  CALL DRWAES(IPX,IPY)
      IPO=IPY
      NC=NC+1
      IDN(NC)=L
200  CONTINUE
1000 CONTINUE
      END
CINTITE
      SUBROUTINE INTITE(XO,YO)
C***** SPECIAL INTERACTIVE TITLE ROUTINE
      DIMENSION IQUICK(30),MSG1(11),MSG2(5),MSG3(5),MSG4(19),MSG5(11)
      DIMENSION MSG6(7)
      DIMENSION IPPX(4),IPPY(4),VTEM(5)
      COMMON/ITITLE/IT(50),JT(50),KT(50),LT(50),MT(50),ST(50),
1  XT(50),YT(50),XU(50),YU(50),XV(50),YV(50),ITC(8,50),NTITE
      COMMON/TEKGPPR/LDEL,LCNT,MAXSR,LTV(17),EN(2),DEL(2),BEG(2),RDX2,
1RDY2,NLINE,NDRAW(50)
      IXTOS(X)=((X-BEG(1))/DEL(1)+600)
      IYTOS(Y)=((Y-BEG(2))/DEL(2)+300)
      XTOSX(X)=(X-600.)*DEL(1)+BEG(1)
      YTOSY(Y)=(Y-300.)*DEL(2)+BEG(2)
      DATA IQUICK/7,0,9,4,0,0,0,2,0,0,
1      0,1,2,6,0,8,5,0,3,0,
2      0,2,0,0,0,0,0,0,0,0/
C***** ANGLE POINT
      DATA MSG1/65,78,71,76,69,32,80,79,73,78,84/
C***** SIZE
      DATA MSG2/83,73,90,69,32/
C***** TITLE
      DATA MSG3/84,73,84,76,69/
C***** ILLEGAL
      DATA MSG6/73,76,76,69,71,65,76/
C***** ENTER IT,ANGLE,SIZE
      DATA MSG4/69,78,84,69,82,32,73,44,74,44,75,44,65,78,71,
1  44,83,73,90/
C***** ENTER TITLE
      DATA MSG5/69,78,84,69,82,32,84,73,84,76,69/
      LCNT=LCNT-50
      CALL PLCHAR(25,39)

```

```

CALL ANMODE
CALL NOTATE(0,LCNT,5,MSG3)
IF(NTITE.EQ.0)GO TO 140
GO TO 20
10 CALL PLCHAR(25,39)
CALL VCURSR(ICAR,XO,YO)
C**** IF NO TITLES THEN ASSUME TO ADD CNE
IF(NTITE.EQ.0)GO TO 140
IF(ICAR.LT.64.OR.ICAR.GE.95) GO TO 210
ICAR=ICAR-64
ICHECK=IQUICK(ICAR)
IF(ICHECK.EQ.0) GO TO 210
GO TO (20,50,70,80,220,140,110,180,300)ICHECK
C**** 7=A ANGLE
C**** 1=C POSITION
C**** 4=D DELETE TITLE
C**** 5=E EXIT FROM INTERACTIVE TITLE
C**** 2=M MOVE TITLE POSITION
C**** 6=N NEW TITLE INSERT
C**** 3=S CHANGE SIZE OF LETTERS
C****
C**** POSITION ON TITLE CLOSEST
20 DSAVE=1.E40
DO 40 K=1,NTITE
XXX=XT(K)
YYY=YI(K)
IF(IT(K).EQ.0)GO TO 30
XXX=XTOSX(XXX)
YYY=YTOSY(YYY)
30 XDX=XXX-XO
YDY=YYY-YO
DIST=XDX*XDX*RDY2+YDY*YDY*RDY2
IF(DIST.GT.DSAVE)GO TO 40
NTF=K
XXP=XXX
YYP=YYY
DSAVE=DIST
40 CONTINUE
CALL POINTA(XXP,YYP)
GO TO 10
C***** MOVE TITLE AND CORRECT ANGLE
50 IF(NTF.EQ.0)GO TO 210
IF(IT(NTF).EQ.0)GO TO 60
XO=IXTOS(XO)
YO=IYTOS(YO)
60 XT(NTF)=XC
YT(NTF)=YO
CALL POINTA(XO,YO)
IF(ICAR.NE.2)GO TO 61
XU(NTF)=XV(NTF)=0.
YU(NTF)=YV(NTF)=0.
GO TO 180
61 IF(ICAR.NE.22)GO TO 62
XU(NTF)=XV(NTF)=YU(NTF)=0.
YV(NTF)=1.
GO TO 180
62 CALL VCURSR(ICA,XO,YO)
XU(NTF)=XO
YU(NTF)=YO
120 CALL POINTA(XO,YO)
CALL VCURSR(ICH,XB,YB)
CALL POINTA(XB,YB)

```



```

      XV(NTF)=XB
      YV(NTF)=YB
C**** PLOT NEW TITLE
      GO TO 180
C****
C**** CHANGE THE SIZE OF THE TITLE
      70 IF(NTF.EQ.0) GO TO 210
         LCNT=LCNT-50
         CALL NOTATE(0,LCNT,5,MSG2)
         CALL ANMODE
         CALL GETIN(1,ST(NTF))
         GO TO 10
C****
C**** DELETE TITLE
      80 IF(NTF.EQ.0)GO TO 20
         IF(NTF.EQ.NTITE)GO TO 100
         IT(NTF)=IT(NTITE)
         JT(NTF)=JT(NTITE)
         KT(NTF)=KT(NTITE)
         ST(NTF)=ST(NTITE)
         XT(NTF)=XT(NTITE)
         YT(NTF)=YT(NTITE)
         XU(NTF)=XU(NTITE)
         YU(NTF)=YU(NTITE)
         XV(NTF)=XV(NTITE)
         YV(NTF)=YV(NTITE)
         DO 90 I=1,8
      90 ITC(I,NTF)=ITC(I,NTITE)
      100 NTITE=NTITE-1
         NTF=0
         GO TO 10
C****
C**** ADD ARROW TO TITLE CODE
C****
      110 NTITE=NTITE+1
         NTF=NTITE
         IT(NTF)=0
         JT(NTF)=0
         KT(NTF)=1
         DO 150 I=1,8
      150 ITC(I,NTITE)=10H
         ITC(1,NTITE)=10HARROW
         GO TO 60
C**** ADD NEW TITLE
      140 LCNT=LCNT-50
         CALL NOTATE(0,LCNT,10,MSG4)
         CALL ANMODE
         CALL GETIN(2,VTEM)
         LCNT=LCNT-50
         CALL NOTATE(0,LCNT,11,MSG5)
         NTITE=NTITE+1
         NTF=NTITE
         IT(NTF)=VTEM(1)
         JT(NTF)=VTEM(2)
         KT(NTF)=0
         ST(NTF)=.15
         CALL ANMODE
         READ (5,170)(ITC(I,NTITE),I=1,8)
      170 FORMAT(8A10)
         GO TO 50
C****
C**** PRINT TITLE AND ASK AGAIN

```

C\*\*\*\*

```

180 IF(NTF.EQ.0)GO TO 210
    I=NTF
    IF(KT(I).EQ.1)GO TO 200
    CALL SWCHAR(1)
    IX=XT(I)
    IY=YT(I)
    IF(IT(I).EQ.1)GO TO 190
    IX=IXTOS(XT(I))
    IY=IYTOS(YT(I))
190 IF(ST(I).LE.0.)ST(I)=.10
    ICH=KIN(ST(I)*.873)
    ICV=ICH*1.6
    IC=JT(I)
    CALL PLCHAR(ICH,ICV)
    DANGX=(XV(I)-XU(I))*SQRT(RDX2)
    DANGY=(YV(I)-YU(I))*SQRT(RDY2)
    ANG=0.
    IF(DANGY.EQ.0..AND.DANGX.EQ.0.) GO TO 195
    ANG=ATAN2(DANGY,DANGX)*57.2957795
195 CALL PTITE(IX,IY,2,ITC(1,I),IC,70,ANG,IPPX,IPPY)
    CALL LINROT(0.)
    CALL SWCHAR(0)
    GO TO 10
200 IA1=IXTOS(XU(I))
    IA2=IYTOS(YU(I))
    IB1=IXTOS(XV(I))
    IB2=IYTOS(YV(I))
    CALL MCVEA(XT(I),YT(I))
    CALL DARROW(IA1,IA2,IB1,IB2)
    GO TO 10

```

C\*\*\*\*

C\*\*\*\* COPY HEADING

C\*\*\*\*

```

300 IF(NTF.EQ.0)GO TO 210
    NTO=NTF
    NTITE=NTITE+1
    NTF=NTITE
    IT(NTF)=IT(NTO)
    JT(NTF)=JT(NTO)
    KT(NTF)=KT(NTO)
    ST(NTF)=ST(NTO)
    DO 310 I=1,8
310 ITC(I,NTF)=ITC(I,NTO)
    GO TO 50
210 LCNT=LCNT-50
    CALL NOTATE(0,LCNT,7,MSG6)
    GO TO 10

```

C\*\*\*\*

C\*\*\*\* END OF INTERACTIVE TEST

220 RETURN

END

CDARROW

```

SUBROUTINE DARROW(IX1,IY1,IX2,IY2)
    CALL DRWAES(IX1,IY1)
    ANG=0.
    IF(IY2.EQ.IY1.AND.IX2.EQ.IX1)GO TO 10
    ANG=ATAN2(FLOAT(IY2-IY1),FLOAT(IX2-IX1))
10 I2=IY2-80.*SIN(ANG-.2)
    I1=IX2-80.*COS(ANG-.2)
    I4=IY2-80.*SIN(ANG+.2)
    I3=IX2-80.*COS(ANG+.2)

```

```

IX=.5*(I1+I3)
IY=.5*(I2+I4)
CALL DRWAES(IX,IY)
CALL DRWAES(I1,I2)
CALL DRWAES(IX2,IY2)
CALL DRWAES(I3,I4)
CALL DRWAES(I1,I2)
RETURN
END

```

CGETIN

GETI0001

```

SUBROUTINE GETIN(NIN,Y)
C
C      SIMULATED MISSION ENGINE TEST CCDE
C      MICHAEL CADDY 3/19/78
C      DIMENSION Y(1),IC(80)
C      FREE FORM INPUT CODE
      NW=0
10 J=0
      READ 20,IC
C      CHECK FOR END OF FILE
      IF(EOF(5).EQ.0)GO TO 30
      NIN=NW
      RETURN
20 FORMAT(80R1)
30 JC=0
      JD=0
      JS=1
      NC=0
      X=0.
40 J=J+1
C      ONLY ONE CARD PER INPUT READ
C      MODIFIED TO READ MORE THAN ONE CARD 4/26/78 MJC
      IF(J.GT.80) GO TO 10
      I=IC(J)
C      CHECK FOR VALID NUMERIC FIELD
      IF(I.GT.32B.AND.I.LT.45B) GO TO 110
C      IGNORE LEAD + SIGN
      IF(I.EQ.45B) GO TO 40
C      SET FLAG FOR NEGATIVE VALUE
      IF(I.NE.46B) GO TO 50
      JS=-1
      GO TO 40
C      CHECK FOR DECIMAL
50 IF(I.NE.57B) GO TO 60
      IF(JC.EQ.-1) GO TO 120
C      IF THIS IS SECOND DECIMAL BLOW OFF TO ERROR CODE
      JC=-1
      GO TO 40
C      CHARACTER IS BLANK TREAT AS COMMA IF NOT LEADING
60 IF(I.EQ.55B)GO TO 80
      IF(I.EQ.56B)GO TO 70
      GO TO 120
70 IF(NC.GT.0)GO TO 90
      NW=NW+1
      IF(NW.GT.NIN) RETURN
      GO TO 40
C      TWO COMMAS ..IGNORE THIS DATA FIELD AND GO ON TO NEXT
80 IF(NC.EQ.0) GO TO 40
C      SHIFT DECIMAL TC NUMBER
90 X=JS*X*10.**JD
      NW=NW+1

```

```

GETI0002
GETI0003
GETI0004
GETI0005
GETI0006
GETI0007
GETI0008
GETI0009
GETI0010
GETI0011
GETI0012
GETI0013
GETI0014
GETI0015
GETI0016
GETI0017
GETI0018
GETI0019
GETI0020
GETI0021
GETI0022
GETI0023
GETI0024
GETI0025
GETI0026
GETI0027
GETI0028
GETI0029
GETI0030
GETI0031
GETI0032
GETI0033
GETI0034
GETI0035
GETI0036
GETI0037
GETI0038
GETI0039
GETI0040
GETI0041
GETI0042
GETI0043
GETI0044
GETI0045
GETI0046
GETI0047
GETI0048
GETI0049
GETI0050
GETI0051

```

```

      Y(NW)=X
      IF(NW.GE.NIN) RETURN
      GO TO 30
110  JD=JD+JC
      NC=NC+1
C     ADD DIGIT TO NUMBER ..CAREFULLY
      X=X*10+(I-33B)
      GO TO 40
C     ERROR CODE
120  DO 130 K=1,80
130  IC(K)=55B
      IC(J)=47B
      PRINT 140 ,IC
140  FORMAT(2X,80R1)
      PRINT 150
150  FORMAT(* EAD FIELD, RE-ENTER DATA*)
      GO TO 10
      END
      SUBROUTINE AXIS(IX,IY,IEY,IEY,ID,IC,AB,DT,EX,IJ,JZZ,NTB,LINE,SIZ)
C
C IX -ORIGIN X-COORDINATE (SCREEN UNITS)
C IY - ORIGIN Y-COORDINATE (SCREEN UNITS)
C IE - AXIS LENGTH (SCREEN UNITS)
C ID - NO. OF DIVISIONS ON AXIS
C IC -- 0=HORIZONTAL LINE ; 1=VERTICAL LINE
C AB--AXIS BEGINNING(REAL UNITS)
C DT--CHANGE IN SCALE BETWEEN TIC MARKS (REAL UNITS)
C NTB--NO. OF MINOR TICKS
C LINE-GRID LINE CCDE(DRAW EVERY N GRIDS)
C SIZ-CHARACTER SIZE
      DIMENSION NODEC(4)
      DIMENSION JA(10)
      JZ=JZZ
      XB=IX
      YB=IY
      CALL SWCHAR(1)
      ICH=KIN(SIZ*.873)
      IE=IEY
      IF(IC.EQ.1) IE=IEY
      ICV=ICH*1.6
      CALL PLCHAR(ICH,ICV)
      CALL CSIZE(JH,JV)
      DATA XY/292./
      DO 1 I=1,4
1  NODEC(I)=0
      Y=0.
C  DETERMINING NO. OF DECIMAL PTS IN LABELS ON AXIS
      CALL ANNOT(IX,Y,Z,JA,IJ,NODEC(1),NO,NODE)
      LL=NO
      CALL ANNOT(IY,Y,Z,JA,IJ,NODEC(2),NO,NODE)
      IF(NO.GT.LL)LL=NO
      CALL ANNOT(AB,Y,Z,JA,IJ,NODEC(3),NO,NODE)
      IF(NO.GT.LL)LL=NO
      CALL ANNOT(DT,Y,Z,JA,IJ,NODEC(4),NO,NODE)
      IF(NO.GT.LL)LL=NO
      NODE=MAX0(NODEC(1),NODEC(2),NODEC(3),NODEC(4))
      EX=0.
C  DETERMINING EXPONENT
      DA=ABS(DT)
      IF(DA) 3,7,3
3  IF(DA-99.) 6,4,4
4  DA=DA/10.

```

GETI0052  
 GETI0053  
 GETI0054  
 GETI0055  
 GETI0056  
 GETI0057  
 GETI0058  
 GETI0059  
 GETI0060  
 GETI0061  
 GETI0062  
 GETI0063  
 GETI0064  
 GETI0065  
 GETI0066  
 GETI0067  
 GETI0068  
 GETI0069

```

      EX=EX+1.
      GO TO 3
5    DA=DA*10.
      EX=EX-1.
6    IF(DA-.01) 5,7,7
7    CONTINUE
      XV=AB*10.**(-EX)
      DA=DT*10.**(-EX)
      AA=IC*2-1
      CH=1-IC
      SH=IC
      NT=ID+1
      JT=JV
      IF(IC.EQ.0) JT=(LL+2)*JH
      NA=IE/ID
C**** JZ= HOW MANY TO SKIP LABELING
60   IF(JT.LT.JZ*NA) GO TO 70
      JZ=2*JZ
      GO TO 60
70   NX=XB+CH*IE
      NY=YB+SH*IE
      CALL MOVABS(NX,NY)
      IDX=-XY*AA*SH*.07
      IDY=XY*AA*CH*.07
C**** NUMBER OF MINOR TICKS
      NTEE=NTB+1
C**** TOTAL NUMBER OF TICKS
      IA=ID*NTEE
      DEL=FLOAT(IE)/FLOAT(IA)
C  LOOP FOR AXIS LINE & TICS
      DELM=FLOAT(IE)/ID
      DO 30 I=1,NT
      NX=XE+DELM*CH*(NT-I)
      NY=YB+DELM*SH*(NT-I)
      KL=2
      DO 30 J=1,NTEE
      CALL DRWABS(NX,NY)
      CALL DRWABS(NX+IDX*KL,NY+IDY*KL)
      CALL DRWABS(NX,NY)
      IF(I.EQ.NT) GO TO 34
      IA=IA-1
      NX=XB+DEL*CH*IA
      NY=YB+DEL*SH*IA
30   KL=1
34   X=IX+IDX
      Y=IY+IDY
      IF(IC.EQ.0) GO TO 61
      X=X-LL*JH
      GO TO 62
61   X=X-JH
      Y=Y-JV
62   XV=AE*10.**(-EX)
      YK=1.
      XN=NX
      CALL ANNOT(XV,X,YK,JA,IJ,J,NO,NCDE)
      NN=IFIX(X)-JH
      JJ=IFIX(Y)
      IF(IC.EQ.0) JJ=JJ-IFIX(.1*JV-IDY)
      NNX=NN-JH*(NO/2)-JH/2
      IF(IC.EQ.1) NNX=IX+IDX-6.0*JH
      IF(IC.EQ.1) JJ=JJ-3*JV/E
      CALL NOTATE(NNX,JJ,5,JA)

```

```

      IN=0
      NTT=NT-1
      DEL=FLOAT(IE)/ID
C   LABEL AXIS
      DO 63 I=1,NTT
      IN=IN+1
      XV=XV+DA
      XN=XN+CH
      YN=YN+SH
      IF(IN.NE.JZ) GO TO 63
      IN=0
      YN=1.
      CALL ANNOT(XV,XN,YN,JA,IJ,J,NC,NODE)
      IF(IC.EQ.0)GO TO 50
      JJ=JJ+(DEL*JZ)
      GO TO 51
50  NN=NN+(DEL*JZ)
      NNX=NN-JH*(NC/2)-JH/2
51  CALL RCTATE(NNX,JJ,5,JA)
63  CONTINUE
      RETURN
      END
      SUBROUTINE ANNOT(X,Y,Z,JA,IJ,N,NO,NCDE)
C   SUBROUTINE TO DETERMINE CHARACTERS FOR LABELLING AXIS
C   OUTPUT GOES IN JA(1-10)   (LEFT JUSTIFIED)
      DIMENSION JA(10)
      XX=X
      I=0
      DO 1 J=1,10
1   JA(J)=32
      ENCODE(10,5,I) XX
5   FORMAT(F10.2)
      CALL KAM2AS(10,I,JA)
      IF(JA(1).NE.32) GO TO 50
      DO 10 I=1,9
      IF(JA(I).NE.32) GO TO 40
10  CONTINUE
      GO TO 50
40  DO 45 J=I,10
45  JA(J-I+1)=JA(J)
      L=12-I
      DO 46 K=L,10
46  JA(K)=32
50  CONTINUE
      IF(Z.GT..5) GO TO 30
      N=0
      DO 110 I=1,10
      IF(JA(I).NE.46) GO TO 110
      IF(JA(I+2).NE.32.AND.JA(I+2).NE.46) GO TO 20
      IF(JA(I+1).NE.32.AND.JA(I+1).NE.46) GO TO 15
110 CONTINUE
      GO TO 25
15  N=1
      GO TO 25
20  N=2
25  CONTINUE
      RETURN
30  CONTINUE
      IF(NCDE.GT.0) GOTO2
      DO 101 I=1,10
      IF(JA(I).NE.46) GOTO101
      J=I-1
      DO 201 K=I,10

```

```

201 JA(K)=32
    GO TC 100
101 CCNTINUE
    GCTC100
    2 DO 12 I=1,10
      IF(JA(I).NE.46) GCTC12
      J=I+NCDE+1
      DO 22 K=J,10
22  JA(K)=32
      J=J-1
      GO TO 100
12  CONTINUE
100 CONTINUE
    NO=5-J
    IF(JA(1).EQ.46.AND.J.NE.1) JA(1)=32
    IF(J.GT.4)RETURN
    DO 461 L=1,J
      JA(6-L)=JA(J+1-L)
461  JA(J-L+1)=32
    RETURN
    END
CAXSCAL
    SUBROUTINE AXSCALE(X,AXL,N,XSTART,XINC,ISIZE)
C *** X-ARRAY OF DATA    AXL-AXIS LENGTH INCHES
C *** N- NO. OF POINTS IN ARRAY $ XINC-INCREMENT PER INCH
C *** XSTART FIRST NO. ON AXIS
C *** ISIZE=0 USE 10 TO THE INCH SCALING ,NOT =0 USE 20
    DIMENSION X(1),L(5)
    XMAX=XMIN=X(1) $ L(1)=1 $ L(2)=2 $ L(3)=5 $ L(4)=10 $ L(5)=20
    IF(AXL.LE.0.)AXL=1.
    IF(ISIZE.NE.0) L(3)=4
    IF(N.LE.1) GO TO 70
    DO 60 I=2,N
      XCHECK=X(I)
      IF(XCHECK.GT.XMAX)XMAX=XCHECK
      IF(XCHECK.LT.XMIN)XMIN=XCHECK
60  CONTINUE
70  F=(XMAX-XMIN)/AXL
    IF(F.EQ.0.) F=XMAX/AXL
    J=-10
    DO 100 I=1,20
      K=10.**J*F
      IF(K.NE.0) GO TC 110
100  J=J+1
110  DO 120 I=2,4
      IF(L(I).GT.K) GO TO 130
120  CONTINUE
130  XINC=L(I-1)/10.**J
      K=XMIN/XINC
      IF((XMIN/XINC).LT.-.06) K=K-1
      XSTART=K*XINC
      XH=(XMAX-XSTART)/XINC
      XL=(XMIN-XSTART)/XINC
      IF(XH.GT.(AXL+.06).OR.XL.LT.-.06) 140,150
140  I=I+1
      GO TC 130
150  CONTINUE
    RETURN
    END

```

What is claimed is:

1. A process for generating a camera-ready hardcopy of a graphical plot of certain physical quantities or existing plots using a general purpose digital computer, a graphics display terminal having an alpha-numeric keyboard and a cursor control, and an associated printer, comprising in sequence the steps of:

storing in the computer an interactive graphics code and coordinate data in tabular form associated with the physical quantities or existing plots, said code being identified by interactive command repetition and feedback for deletion, addition and transition of lines, points and labels while processing the stored coordinate data;

plotting on the screen the stored coordinate data in a format according to said code;

generating on the screen lines between the plotted coordinate data according to said code;

revising the coordinate data and the lines plotted on the screen by manipulating the cursor and keyboard in accordance with said code, said revising step further including

adding coordinate data to create new lines, deleting coordinate data to remove lines, and moving coordinate data to change the position of existing lines; and

printing a hardcopy of the revised plot on the printer.

2. A process as recited in claim 1, wherein, immediately following said step of revising the coordinate data, the following step is included:

5 labeling appropriate areas of the revised plot on the screen by manipulating the cursor and keying the computer in accordance with the interactive code.

3. A process as recited in claim 2, wherein immediately following said step of labeling, the following step

10 is included: shading appropriate areas of the revised plot on the screen by manipulating the cursor and keying the computer in accordance with the interactive code.

15 4. A process as recited in claim 3, wherein immediately following said step of shading, the following step is included:

applying appropriate reference symbols and characters to the revised plot on the screen by manipulating the cursor and keying the computer in accordance with the interactive code.

5. A process as recited in claim 1, wherein said step of plotting comprises:

25 establishing appropriately scaled and titled coordinate axes on the screen; and establishing appropriate grid lines substantially horizontally and vertically on the screen.

\* \* \* \* \*